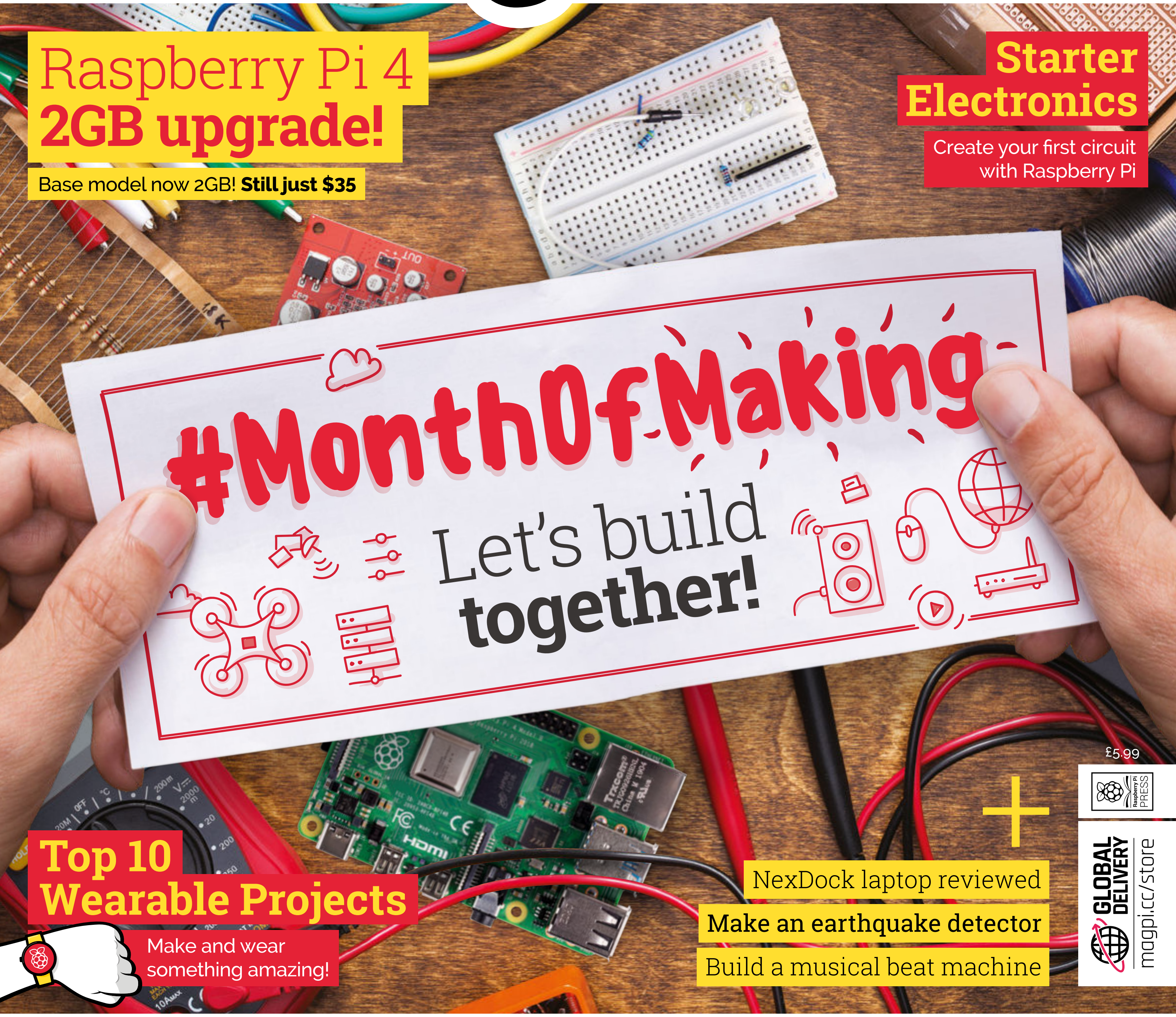
USA OFFER! 12 ISSUES NOW $60 MAGPI.CC/USA

The MagPi

Issue 91 | March 2020 | magpi.cc

The official Raspberry Pi magazine



Raspberry Pi 4 2GB upgrade!

Base model now 2GB! **Still just $35**

Starter Electronics

Create your first circuit with Raspberry Pi

Top 10 Wearable Projects

Make and wear something amazing!

+

NexDock laptop reviewed

Make an earthquake detector

Build a musical beat machine

£5.99

GLOBAL DELIVERY magpi.cc/store

40 PAGES OF PROJECTS & TUTORIALS

Over 10,000 Different Boards and Modules In Stock

NO ONE KNOWS BOARDS LIKE DIGI-KEY!

- Open Source Community Boards
- Evaluation and Development Boards
- Wireless Modules
- Single Board Computers
- Reference Designs
- Programmers, Emulators and Debuggers

DESIGNS START HERE!
DIGIKEY.COM/BOARDS

Digi-Key® ELECTRONICS

MAKER.IO POWERED BY DIGI-KEY **#MAKEWITHDIGIKEY**

ECIA MEMBER Supporting The Authorized Channel

WELCOME to *The MagPi* 91

***The MagPi* ran its first #MonthOfMaking last March almost on a whim.** We had lots of cool ideas for projects, and wanted to get on with them. Then Rob had the bright idea of doing a maker special all month: a time where we would all commit to getting on with that thing we'd been planning for months.

It was a rip-roaring success, and we've been looking forward to this month ever since. So #MonthOfMaking is back (page 32). If you're on Twitter, make sure you search for the hashtag (and our other one: #MyLatestBuild) and get in touch with other *The MagPi* makers.

This month I made a Raspberry Shake (page 42) and installed it via power-line networking in my conservatory. There my little earthquake-detector sits, 24-hours-a-day, quietly reporting on seismic activity to other citizen scientists around the world.

Of course, the real joy of making is (for me) learning something new: in this case how a geophone sensor works. I'm also going to get some practice in working with the datasets. But I'm just happy to make something for the sheer joy of seeing it exist in the world.

Lucy Hattersley Editor

EDITOR
Lucy Hattersley

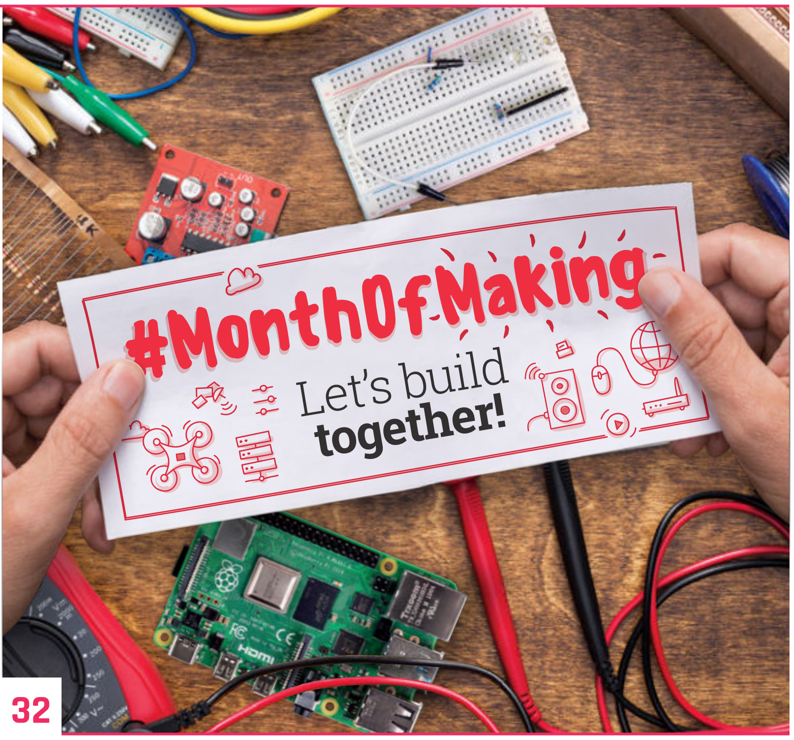
Lucy is editor of *The MagPi* magazine. She's just discovered how much her Game Boy Micro is worth, while holding a screwdriver in one hand and an Adafruit 1.3-inch bonnet in the other. Thanks, Twitter.

magpi.cc

GET A RASPBERRY ZERO W KIT
WITH A SUBSCRIPTION!
PAGE 30

Contents

> Issue 91 > March 2020



32

20

Reachy

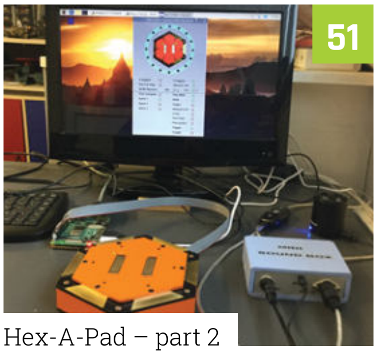
14

NeoPixel LED Mirror

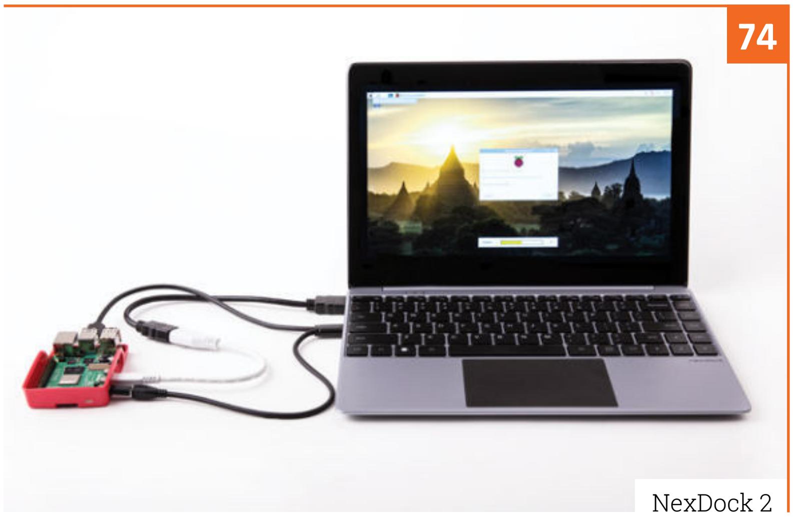
42

Raspberry Shake

51

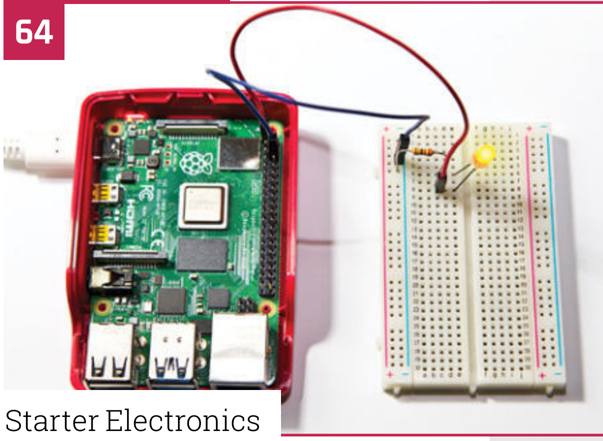

Hex-A-Pad – part 2

74


NexDock 2

82

Ben Nuttall interview


Starter Electronics

WIN

A SIGNED RASPBERRY PI

DESKTOP KIT

95

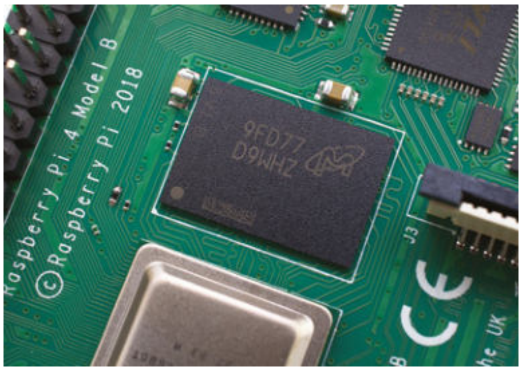
Raspberry Pi 4 now comes with 2GB RAM minimum

Say hello to the new entry point for the family: Raspberry Pi 4 comes with 2GB (and we say farewell to the 1GB model). By **Gareth Halfacree**

When Raspberry Pi 4 launched, it came with the family's first choice of RAM (random-access memory) capacities: 1GB, 2GB, and 4GB. Thanks to falling RAM prices, Raspberry Pi 4 with 1GB is being retired; Raspberry Pi 2GB is now the new entry point to the family.

The Raspberry P4 with 2GB has had a price reduction: it now matches the $35 recommended retail price of the former 1GB model. Entry-level newcomers get double the amount of RAM at no extra cost. Moving from 1GB to 2GB offers a lot more headroom: the additional RAM makes using Raspberry Pi considerably smoother, especially for multitaskers who like to keep several programs running at the same time.

"The additional RAM makes using Raspberry Pi considerably smoother"



Raspberry Pi 4 with 2GB and 4GB models are identical designs, bar the memory module used, and fully compatible with each other's software and accessories

Those using multiple high-resolution displays can dedicate more memory to the graphics processor while keeping the CPU fed with RAM.

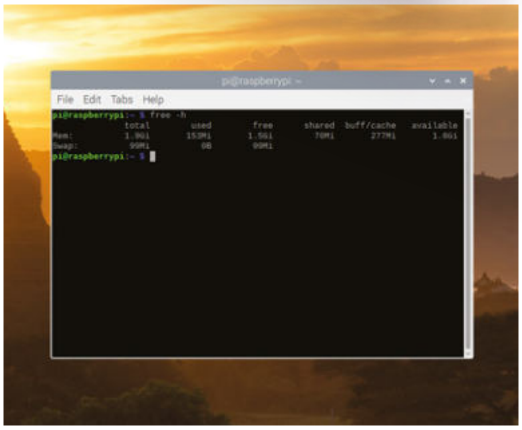
Raspberry Pi 4's line-up now comprises just two models: the 2GB, suitable for most users; and the 4GB, better for power users and heavy multitaskers.

How much RAM do I have?

The amount of RAM each Raspberry Pi 4 model has is printed on the box; once it's set up, though, it can be a little harder to tell. For the lowdown on exactly how much RAM you have – and how much you are using – open a Terminal and type:

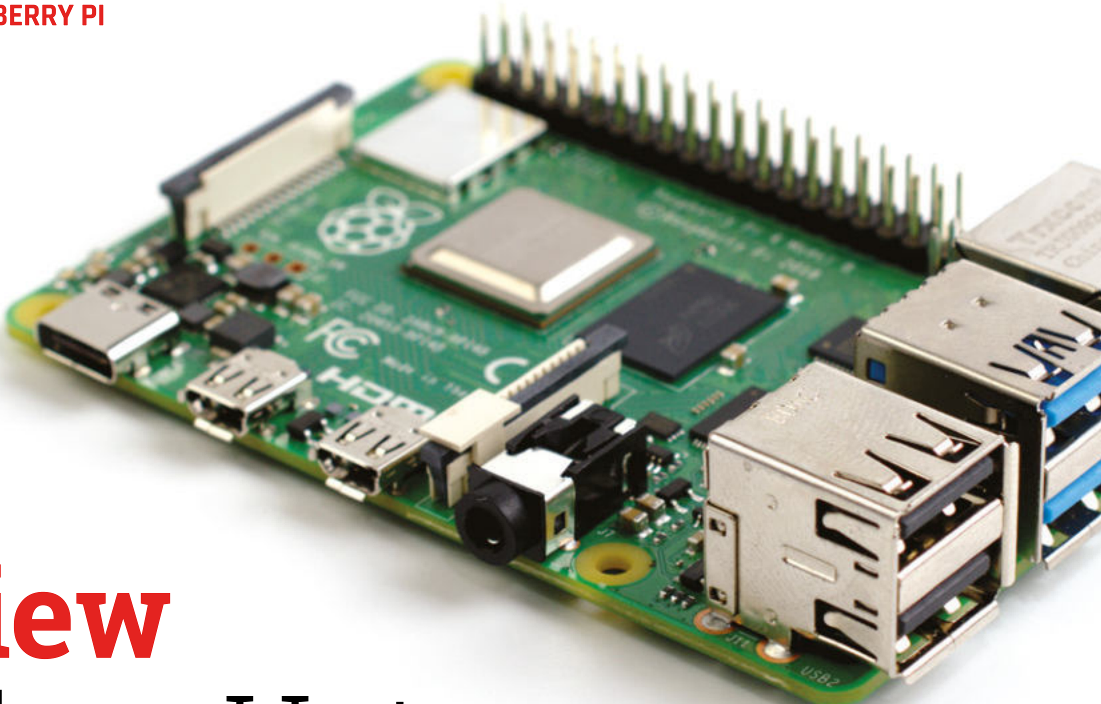
```
free -h
```

The 'total' column is the memory Raspberry Pi has, minus that reserved for the GPU; 'used' is how much is actively used; 'free' is how much is available to use; 'shared' is an obsolete statistic, kept for historical reasons; 'buff/cache' is the amount of memory used by buffers and caches; and 'available' is the amount of memory which could be allocated if these were discarded. Finally, the 'swap' row is the amount of older data which has been moved out of RAM into a swap file on Raspberry Pi's storage device to free up extra memory.



```
pi@raspberrypi:~ $ free -h
              total        used        free      shared  buff/cache   available
Mem:          1.9Gi       153Mi       1.5Gi        70Mi       277Mi       1.6Gi
Swap:          99Mi          0B        99Mi
pi@raspberrypi:~ $
```

▲ Enter a simple Terminal command to find out the total RAM of your Raspberry Pi, and how much of it is free



Interview with Eben Upton

Eben discusses the retirement of Raspberry Pi 4 with 1GB, and the amazing growth Raspberry Pi has seen while maintaining its signature $35 price point

▲ Eben Upton has seen Raspberry Pi increase in performance by an order of magnitude since the original launch eight years ago

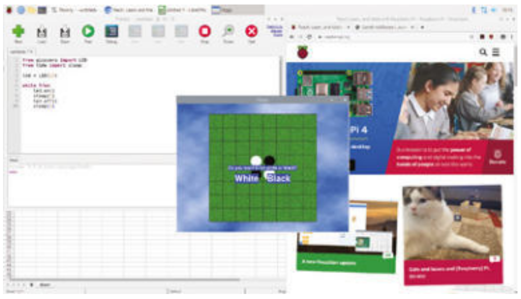
"We moved to 2GB as the entry point for Raspberry Pi 4 because memory prices have come down, and we can afford to," explains Eben Upton, founder of the Raspberry Pi Foundation.

This market shift has allowed a doubling of the RAM without an increase in price. "We couldn't afford it when we launched Raspberry Pi 4. We've had to breathe in quite a long way to make it work at $35, but it's really important because that's what we do.

"If you look at the past eight years," Eben continues, referring to the original launch of Raspberry Pi Model B with its single-core 700MHz processor and just 256MB of RAM, "you've now got eight times as much memory, you've got about 40 times as much processing power, about ten times as much input/output bandwidth. You've got four times as many pixels on-screen and you've got two screens, and you've added dual-band WiFi and Bluetooth – and your $35 from 2012 is about $40 now, accounting for inflation, so you've kind of got a five-dollar real-terms price cut as well.

"It was really important to us to keep pushing the envelope in terms of what's doable – that's the story behind the move to 2GB: trying to make sure we keep pushing forward so we have the best possible desktop experience at the signature price point. 2GB is a much more viable desktop platform than 1GB; 1GB is great for embedded, but for a desktop platform it's just a little bit too tight. So what it means is we're now back to having a really, really viable desktop machine at our signature price point."

In a world where desktops are frequently equipped with 4GB, 8GB, or even 16GB of RAM, there's a reason Raspberry Pi performs so well, even with just 2GB: frugality. "If you look at Windows, or even a traditional Linux desktop distro, there's been a sort of relaxation," says Eben. "As there's been more memory available, people have loosened their belts a little bit and sort of flumped down and started consuming more memory, when we really haven't. We're still using an LXDE-derived desktop environment; you know, we care about every 10MB of memory usage. That's the reason why the 2GB model is a really, really useful desktop."



▶ Moving to 2GB makes Raspberry Pi 4 much more responsive during multitasking – running several applications at the same time

Code Club celebrates Kenyan success

After winning over community leaders, Code Club and Kids Comp Camp are inspiring young coders in rural Kenya, reports **Rosie Hattersley**

▲ Pupils at Kabuku Primary School in Kenya enjoy weekly Code Club sessions

A school in Kenya with no prior computing experience has won over an initially sceptical community, and now runs a thriving Code Club.

The Code Club at Kabuku Primary School, two hours north of Nairobi, is run by Lena. Lena works for an education-focused charity, partnered by Code Club in Kenya, called Kids Comp Camp (**kidscompcamp.com**). Its aim is to improve digital literacy and computing skills in rural parts of the country.

Kids Comp Camp approached the state-run school about setting up a Code Club, having discovered that computing was not part of the curriculum there or in neighbouring schools.

The first hurdle was convincing the local community and Kabuku's headteacher that devoting time to computing would be of value, compared to the practical skills they saw as critical to the children's future prospects. Six months of presentations and meetings followed.

Next, they had to source the hardware needed. Kids Comp Camp donated 15 Raspberry Pi 2 computers. Necessity being the mother of invention, these were connected to monitors with chicken wire!

" Attendees have no existing computing knowledge, so first have to master basic navigation skills "

▲ With no existing computing curriculum, this stone building became their computer lab

◀ The would-be coders first have to learn the basics of how to use a computer. Coding projects are provided via PDF

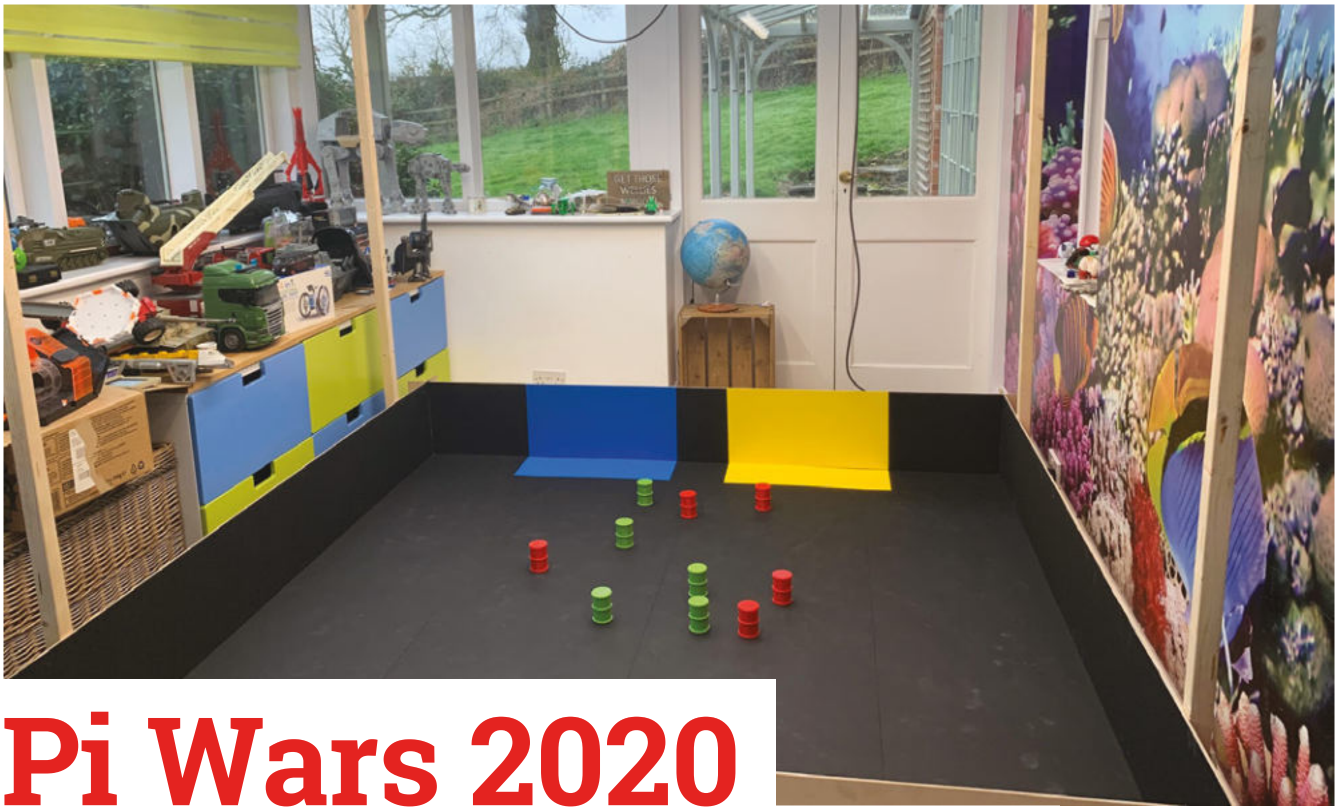
Kabuku's Code Club, based in the stone building that is now the school's computer lab, runs once a week for two hours. Attendees have no existing computing knowledge, so first have to master basic navigation skills before diving into the world of coding.

Rapid progress

When Code Club's International Programme Manager James Aslett visited Kabuku Primary School's Code Club, he was delighted to find just how readily members took to coding. "Children who hadn't ever used a computer six months ago confidently talked through their ideas and decisions with code," he reports.

However, isolated success stories are not enough. Educating communities where computing experience is minimal is critical. "We need passionate and knowledgeable people advocating for the relevance of computing at a local level. There is also a need for engaging resources that excite young people and help them make the most of their hardware," says James.

If you're in the UK, USA, or the Republic of Ireland, head to **codeclub.org** to find out how you can get involved with Code Club in your community. If you're based in the rest of the world, visit **codeclubworld.org** to learn more.



Pi Wars 2020 robots raring to go

Glorious chaos awaits as organisers declare Pi Wars 2020 is going to be a disaster. Enjoy the show, says **Rosie Hattersley**

▲ Catastrophe and chaos will proliferate at this year's Disaster Zone-themed Pi Wars. This is the new Eco Disaster course

▼ Scarab won the Beginner's league in Pi Wars 2020

Pi Wars 2020 is going to be an absolute train wreck – at least that's the hope of its organisers, Mike Horne and Tim Richardson. With a somewhat apocalyptic atmosphere swirling when it came time to choose a theme, Pi Wars organisers declared the 2020 event would have a Disaster Zone theme.

" Fans of zombie films, dystopia, and event horizons rubbed their hands in glee "

Fans of zombie films, dystopia, and event horizons rubbed their hands in glee. The most switched-on 76 teams (of the 128 that applied) nabbed a place in the three-day competition which sees competitor Raspberry Pi-controlled robots pit their skills against each other in a range of non-destructive battles and challenges. Both autonomous and remote-controlled robots jostle for victory by completing up to seven fiendishly complex tasks.

DIY designs

Newbies, veterans, and school teams each have dedicated competition days, helping to ensure everyone has a fair shot of victory. Teams from 17 countries are taking part. Unlike TV's *Robot Wars* (the original inspiration), there's no celebrity version, and each team is expected to design, build, and test their own robot. Competitor entries to this year's event, held over the final weekend of March at the University of Cambridge Computer Laboratory, filled up months ago, and teams have been blogging about their robot's build progress: **magpi.cc/piwars2020blogs**.

Spectator tickets are available from **magpi.cc/piwars2020**.

Universal & Unique

UniPiCase

for Raspberry Pi 4

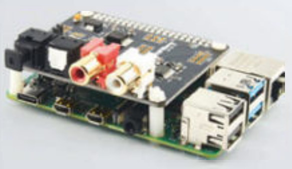
Perfect option for your DIY or high-volume commercial projects

Fits all kinds of HATs
Ideal PoE HAT case

Use with HAT or Pi alone
Simple and professional

Rapid, tool-free assembly
GPIO cable pass-through

Two wall mount options
Well ventilated



www.UniPiCase.com

Designed and manufactured in North America

NeoPixel LED Mirror

MAKER

Alex Schepelmann

A DIY YouTuber who dabbles in Raspberry Pi, robotics, electronics, and 3D printing. During the day, he's an engineer working on robotics and computational modelling.

magpi.cc/supermake

Your face in lights thanks to 576 LEDs. **Rob Zwetsloot** puts on his best hat and takes a look at this 'mirror', and himself

Magic mirrors seem to now be a rite of passage for many makers and Raspberry Pi aficionados. It's a fun project, but we think this LED mirror from Alex Schepelmann is a little more striking.

"The project uses a Raspberry Pi 3B+, a Raspberry Pi Camera [Module], Python, 3D printing, and 576 NeoPixel LEDs to create an interactive art piece that shows you your reflection in 'low resolution' by lighting up a grid of LEDs," says Alex.

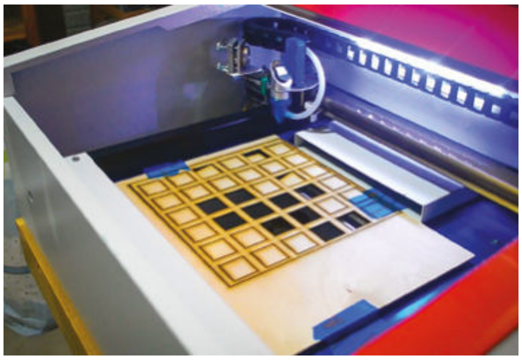
In essence, it's taking your picture using a Raspberry Pi Camera Module, converting it to a low-resolution picture, and then setting the LEDs to the same colour as the individual pixels in the resulting image. Magic? Yes. Practical? No. Fun? Absolutely.

Mirror art

Where did such an idea come from, though? "I was inspired by the various 'analogue mirrors' made by Daniel Rozin," Alex reveals. "The Children's Museum of Pittsburgh, where I built an exhibit for a Systems Engineer class that I took during graduate school, had one of Mr Rozin's mirrors on display. The mirror at the Children's Museum used blocks of wood and servo motors to display images of people who were standing in front of it in low resolution. Ever since then, I've been following Daniel's work, and wanted to build one of his mirrors myself. I thought that such a project would be perfect for my YouTube channel (**magpi.cc/supermake**), because it would allow me to put my own twist on the concept while simultaneously teaching people about programming, 3D printing, laser cutting, and more!"

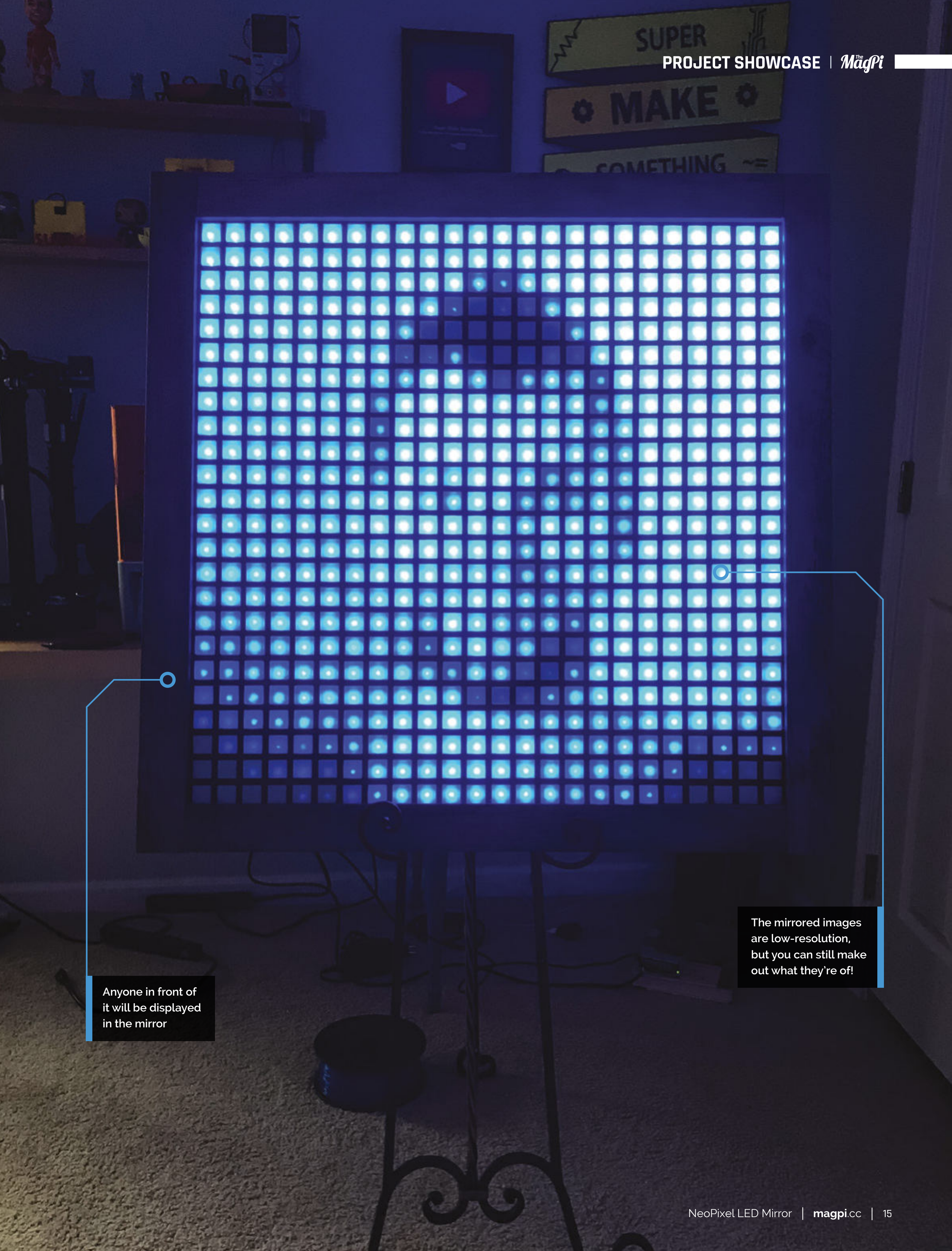
The build itself has an impressive list of components. Alex created a custom prototype

> Shows you your reflection in low resolution by lighting up a grid of LEDs



▲ Alex eventually got a laser cutter to help speed up production – like for these mounting grids

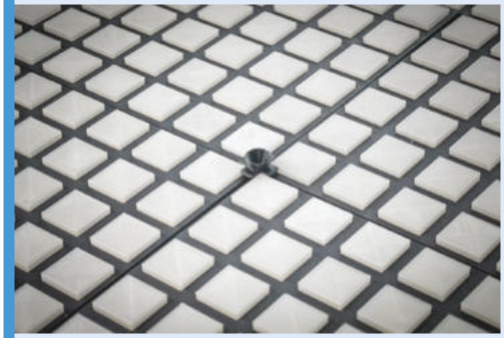
▼ A Raspberry Pi handles it all, although another computer is on hand for project info



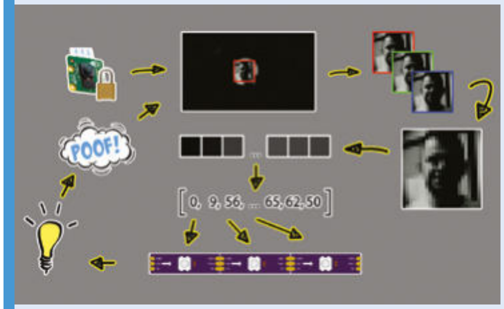
Anyone in front of it will be displayed in the mirror

The mirrored images are low-resolution, but you can still make out what they're of!

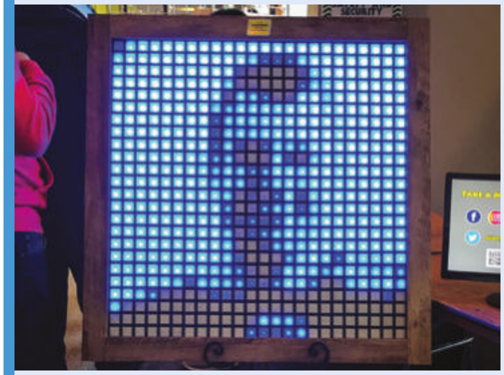
Mirroring with LED



01 Camera settings are locked as the code starts, before capturing the image and selecting a small region of 24×24 pixels.



02 This image is then converted to greyscale, after which the code extracts one of the image's colour planes from the image as an array. This array contains the brightness information for each pixel of the extracted 24×24 region. This square array is then reshaped into a 1×576 vector, and brightness values are assigned to LEDs.



03 Brightness values are used to light up each pixel, after which the image is cleared and the image capture/display process are repeated. In order to be able to display images as quickly as possible, the Python code is optimised to operate on vectors and to minimise the number of for loops.

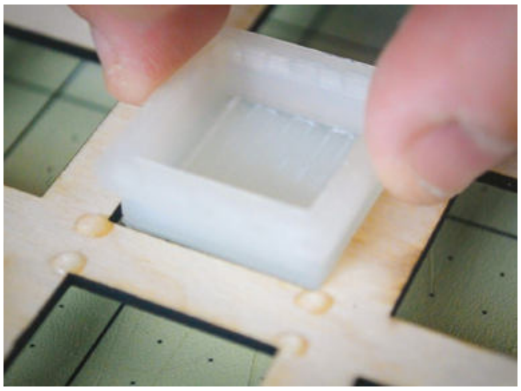
▲ People were taking selfies of themselves with their 'reflection'

PCB, 3D-printed and laser-cut several parts, and connected 24 strips of 24 LEDs to make the magic 576 number. A Raspberry Pi was used to power it due to its size, ability to run Python and address all the LEDs, along with the Raspberry Pi Camera Module which makes it all possible.

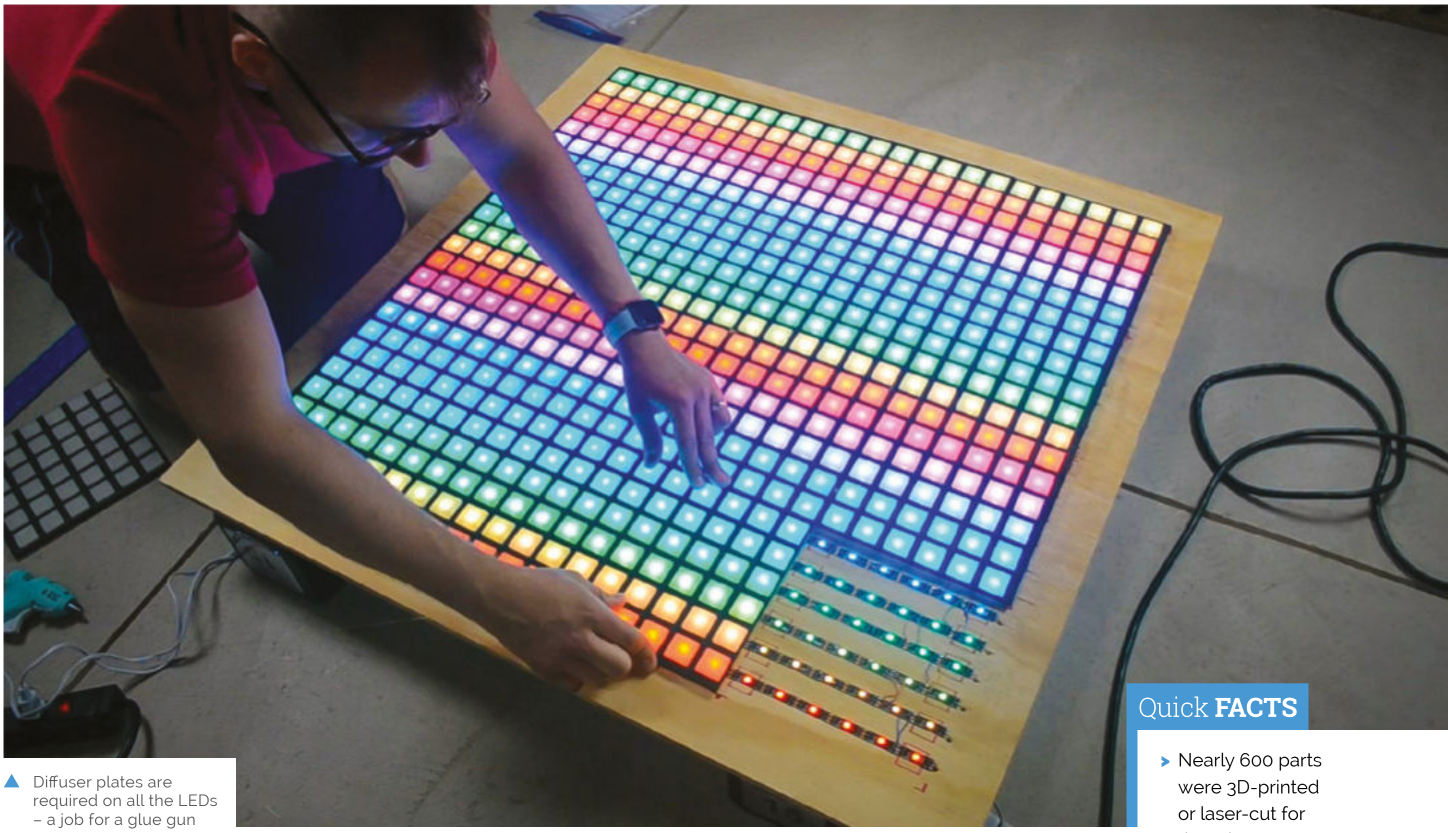
On display

With such an unconventional project, you might expect some issues when it was finally unveiled. However, it went down very well.

"The project made its debut at the 2019 Cleveland Maker Faire, where it ran for over eight hours during the event without a single hiccup," says Alex. "An advantage of being able to run everything via Python code is that I could adjust camera settings on the fly based on lighting conditions in the location where I was at,



▶ Each LED cover was 3D-printed out of clear PLA and then glued into a laser-cut mounting grid


▲ Diffuser plates are required on all the LEDs – a job for a glue gun

Quick FACTS

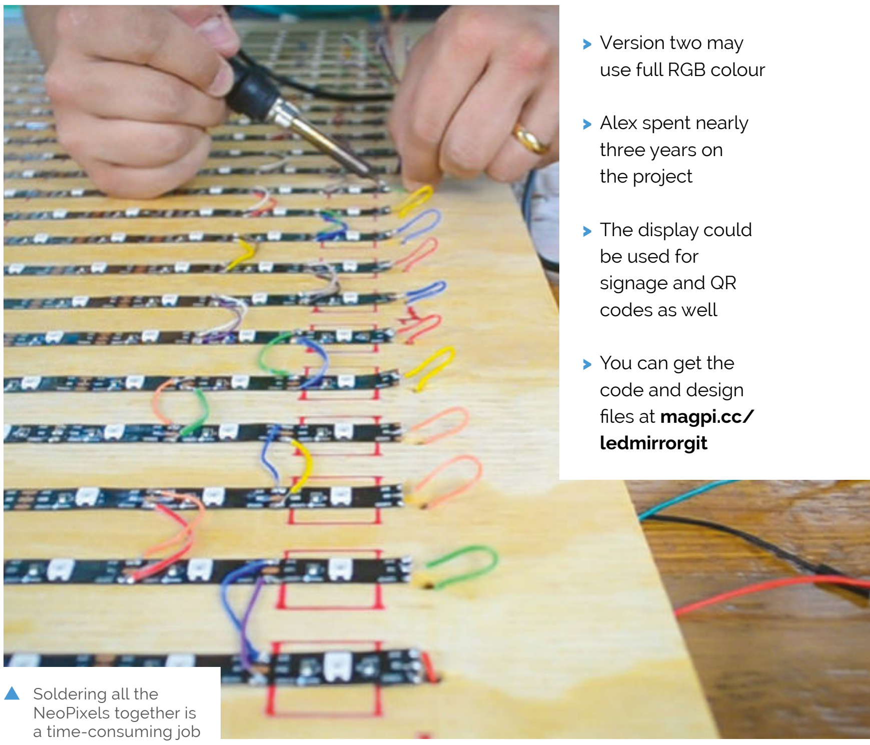
- Nearly 600 parts were 3D-printed or laser-cut for the mirror
- Version two may use full RGB colour
- Alex spent nearly three years on the project
- The display could be used for signage and QR codes as well
- You can get the code and design files at **magpi.cc/ledmirrorgit**

" People were fascinated by the mirror and enjoyed moving their limbs and making faces in front of it "

making sure that the mirror clearly displayed the reflections of visitors throughout the day.

Maker Faire attendees interacted with the mirror and stopped by the Super Make Something booth to learn more about the YouTube channel, Raspberry Pi, and Python programming. "One of my favourite observations I made during this event is that the mirror captured the interest of an audience with a broad age range – people between the ages of 5 and 65 were fascinated by the mirror and enjoyed moving their limbs and making faces in front of it, excited to see what would happen."

If you've not managed to see the mirror in person, all is not lost. Alex has been in discussions to add the mirror to the Great Lakes Science Center, very hopefully with upgrades. Look out for more info on his YouTube channel.


▲ Soldering all the NeoPixels together is a time-consuming job

AdventurePi

Zach Levine's portable arcade project lets you choose your own AdventurePi, as **David Crookes** explains

MAKER

Zach Levine

Zach is a programmer, designer, tinkerer, YouTuber, and howchoo founder living in Tampa, Florida.

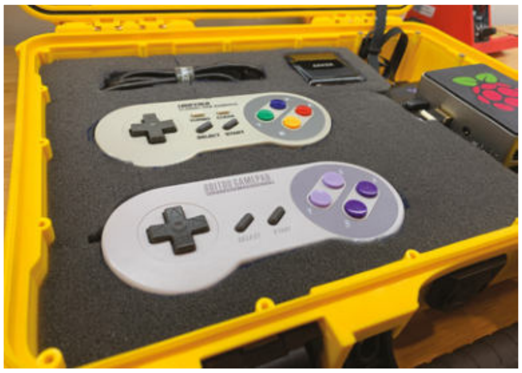
magpi.cc/adventurepi

As big fans of gaming, Zach Levine and his brother ended up owning a good number of video game consoles, including the Game Boy, Virtual Boy, NES, Nintendo 64, and Sega Saturn. "But we also frequented the arcade at the local bowling alley quite often," Zach tells us, and this love of coin-operated gaming machines never left him.

As such, when pondering his next project with a Raspberry Pi, he decided to create a portable gaming device that would be as robust as an arcade machine. "A lot of people my age move around a lot or live in apartments and no one really wants to lug around a free-standing cabinet, so I looked to create a device that would feature full arcade controls."

In fact, Zach ended up creating two different versions of his project. As well as an Arcade Edition that features a removable panel containing arcade buttons and a removable joystick, he produced a Console Edition that uses a foam block insert with cutouts for controllers and accessories.

"My primary goal was to make the project as accessible as possible and introduce more people to Raspberry Pi, hobby electronics, and retro gaming," he says, opting early on to use the RetroPie OS (**retropie.org.uk**) which allows games originally made for older systems to be enjoyed with minimal fuss via emulation.



The Console Edition is essentially a Raspberry Pi with RetroPie installed connected to a display in a case with foam inserts for controllers and accessories

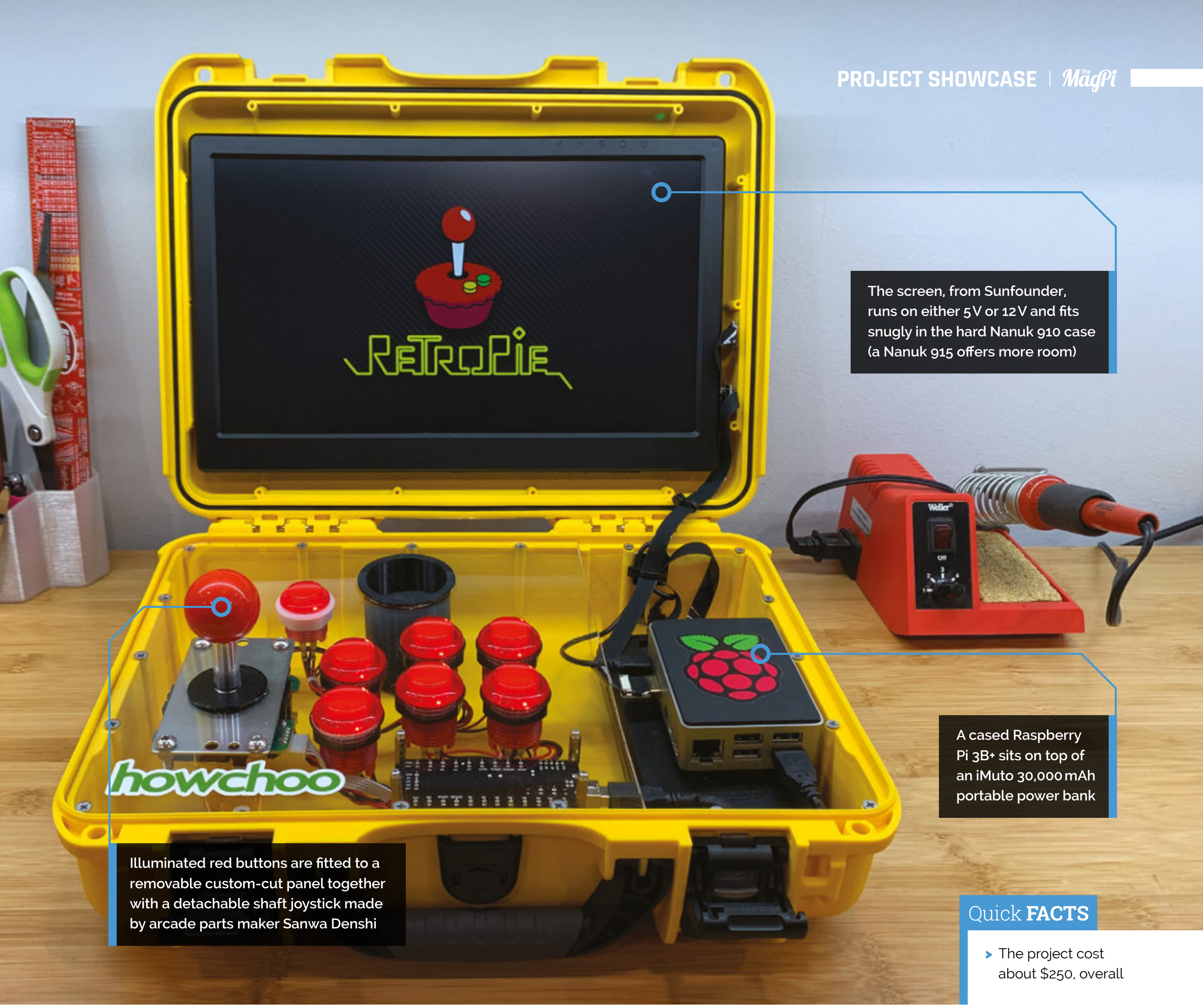
Hardcore gaming

Since RetroPie does not yet work on a Raspberry Pi 4, Zach had to use the 3B+ model instead. Planning proved to be the trickiest part of the setup, but he knew he needed a display that could run on either 12 V or 5 V, and a single power supply that would output enough amperage to run the screen and Raspberry Pi simultaneously for hours.

"No one really wants to lug around a free-standing cabinet"

"From there, it was a matter of finding a sturdy case that had a removable panel for the Arcade Edition and a foam insert for the Console Edition," he says. A Nanuk 910 waterproof hard case worked well, requiring only a little bit of shaving to the inside with a box cutter so that the screen would fit nicely in the lid, while allowing room for the wires.

To avoid the controller smashing into the screen when the case is closed, the joystick unscrews for easy storing



The screen, from Sunfounder, runs on either 5 V or 12 V and fits snugly in the hard Nanuk 910 case (a Nanuk 915 offers more room)

A cased Raspberry Pi 3B+ sits on top of an iMuto 30,000 mAh portable power bank

Illuminated red buttons are fitted to a removable custom-cut panel together with a detachable shaft joystick made by arcade parts maker Sanwa Denshi

Quick FACTS

- The project cost about $250, overall
- There are two versions: Arcade and Console Editions
- It displays to a 13.3-inch screen
- It uses RetroPie for emulation
- Strong nylon fasteners secure the components

"I found some slim 90 degree cables normally used on drones, because there wasn't the space for normal micro USB and HDMI cables," he says.

Joystick joy

For the Arcade Edition, Zach had to essentially build a USB controller. "I made use of illuminated buttons and a joystick connected to a small PCB that turned the setup into a USB device," he explains. He took inspiration from Pimoroni's Picade cabinet. "The controls use the same layout because I figured they'd worked things out already."

Even so, Zach encountered a problem that almost killed the project before it started: how to add a joystick within a closable case. "I messed around with a few hinging mechanisms and then found a company that makes a removable joystick," he says of the fortunate solution.

As for the Console Edition, that has been a case of cutting the foam to house controllers, chargers, wires, and other gaming peripherals that can connect to the Raspberry Pi, but the fun isn't stopping there. Zach believes AdventurePi will eventually be developed as a much smaller version using a custom power supply. "I'd also like to add a power button that automatically shuts [Raspberry] Pi down when the case is closed," he says.

Slim micro USB and HDMI FPV cables were needed because of the small amount of space between the display ports and the case

Reachy

Meet the expressive and flexible open-source robot powered by a Raspberry Pi. **David Crookes** reaches out

MAKER

Pierre Rouanet

Pierre Rouanet studied human-robot interaction and AI for eight years in the INRIA Flowers research lab, where he obtained his PhD. He co-founded Pollen Robotics to imagine new robotic creatures.

pollen-robotics.com

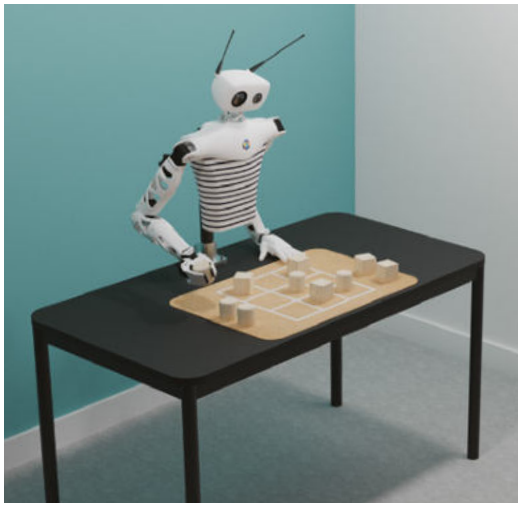
Say hello to Reachy – a humanoid robot that you're likely to love to bits when you first set eyes on it. Blessed with futuristic styling and able to carry out many tasks with precision, it can interact with the environment and with us humans, making for a rather endearing creation.

Which part of Reachy you'll love, however, will mostly depend on the configuration you decide to buy – assuming you have enough money, given that the prices start at €9990.

The basic model, for instance, comes with just a torso and one arm, while 'expressive' adds a Johnny Five-like head. An advanced option gives Reachy an extra arm, but in each case there's a fine heart beating inside: a Raspberry Pi 4 running the Raspbian operating system.

Ideal choice

According to Pierre Rouanet, co-founder and CTO of Reachy creator Pollen Robotics, the decision to use Raspberry Pi 4 came after much debate. "We wanted to provide a simple and well-known setup with a supportive community that would let our users quickly understand, adapt, and modify the basic tool we were providing. Raspberry Pi has always been a very good solution for this."



Reachy played tic-tac-toe against humans at CES 2020, running entirely on Raspberry Pi. Reachy would image-analyse the board, recognise the pawns, and use simple AI to choose what to play next. It would use higher-level control to grasp a pawn and place it

Reachy is open-source and developers can program it using Python, which opens up the possibilities of what it can potentially do. Indeed, Pollen Robotics initially created the robot to help researchers study arm-control in humans, but it's evolved a lot since.

"When we started Reachy in 2016, our former researcher colleagues had wanted to see how an amputee could easily control a prosthetic arm, and they needed something that could closely reproduce human motion and shape," Pierre says. "But we developed new features, including using machine learning for control. We also wanted to work on its ease-of-use to extend the range of its potential users."

" We developed new features, including using machine learning for control "

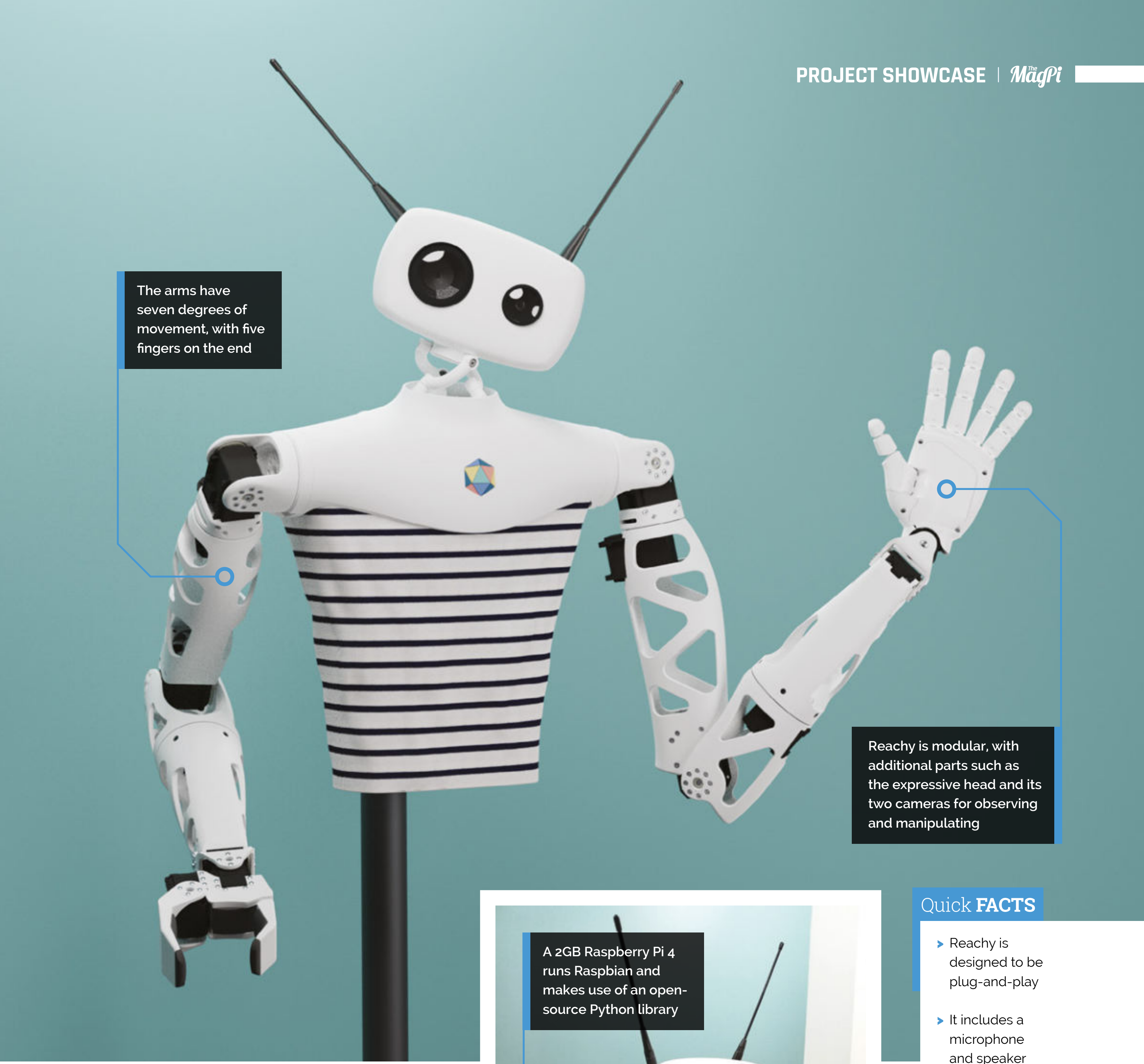
To that end, Pollen Robotics has pre-installed its own Python API and some extra tools for communicating with all the motors and sensors via USB-to-serial communication.

"[Raspberry] Pi is actually running the whole synchronisation loop that retrieves all of the sensors values, and it publishes new commands for the effector (it runs at ~100Hz for Reachy, which is higher than most synchronisation loops in humans)," says Pierre. "On top of that, we run a higher-level application."

Machine learning

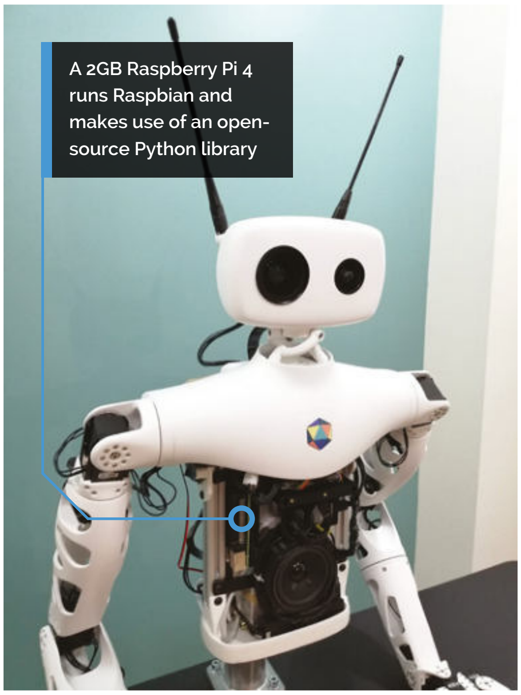
Key to the robot is its built-in artificial intelligence. "We wanted to provide high-end and efficient tools for machine learning," Pierre continues. "Our users require lots of power to perform analysis from the sensors, such as live object recognition and tracking, voice recognition, complex trajectory generation, and so on."

As luck would have it, work on Reachy coincided with the arrival of the Google Coral AI accelerator



The arms have seven degrees of movement, with five fingers on the end

Reachy is modular, with additional parts such as the expressive head and its two cameras for observing and manipulating



A 2GB Raspberry Pi 4 runs Raspbian and makes use of an open-source Python library

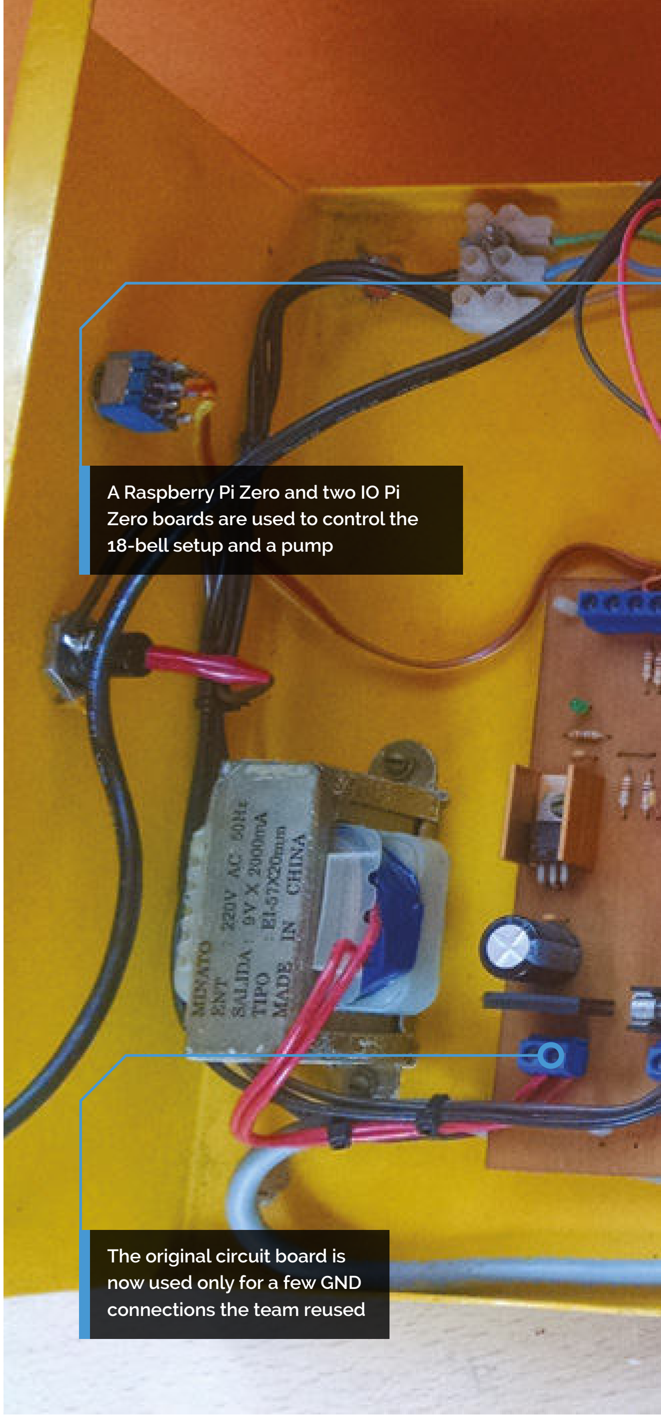
and the new USB 3 ports in Raspberry Pi 4. "It was perfect timing," says Pierre. "We could run all the machine learning we needed, while still providing a simple ready-to-use setup and on top of that, we don't need to rely on a cloud service."

Pierre says Reachy currently shines best when it is manipulating simple objects and interacting with humans. As such, it's primarily intended for use in food and customer service, research, and development. But Pollen Robotics envisages a lower-cost version for hobbyists at some stage which would make for a rather exciting development for the Raspberry Pi community. "This is definitely something that I would like to encourage and see emerge," Pierre says.

Quick **FACTS**

- Reachy is designed to be plug-and-play
- It includes a microphone and speaker
- The innards are covered by fabric
- Google's Coral AI accelerator is also inside
- Only a handful are initially being made

◀ The software, design of all 3D parts, and the data used to pre-train Reachy on some tasks are all open-source



A Raspberry Pi Zero and two IO Pi Zero boards are used to control the 18-bell setup and a pump

The original circuit board is now used only for a few GND connections the team reused

El Carrillon

Tired of hearing the same tune year after year, two makers hacked the bells of an enormous tower to ring the changes. **Rosie Hattersley** hears how

MAKER

Mariano Martinez Peck & Gerardo Richarte

Buenos Aires-based systems engineer and instantiations software specialist Mariano enjoys Raspberry Pi-based IoT projects. Gerardo founded satellite and geospatial solutions company Satellogic. Raspberry Pi and Arduino reignited his love of electronics.

@MartinezPeck
satellogic.com

Most Raspberry Pi projects we feature debut privately and with little fanfare – at least until they're shared by us.

The El Carrillon project, however, could hardly have made a more public entrance. In September 2019 it was a focal point of Argentina's 49th annual Fiesta Nacional de la Flor (National Flower Festival), where its newly overhauled bell tower proudly rang out a brand-new, Raspberry Pi-enabled tune.

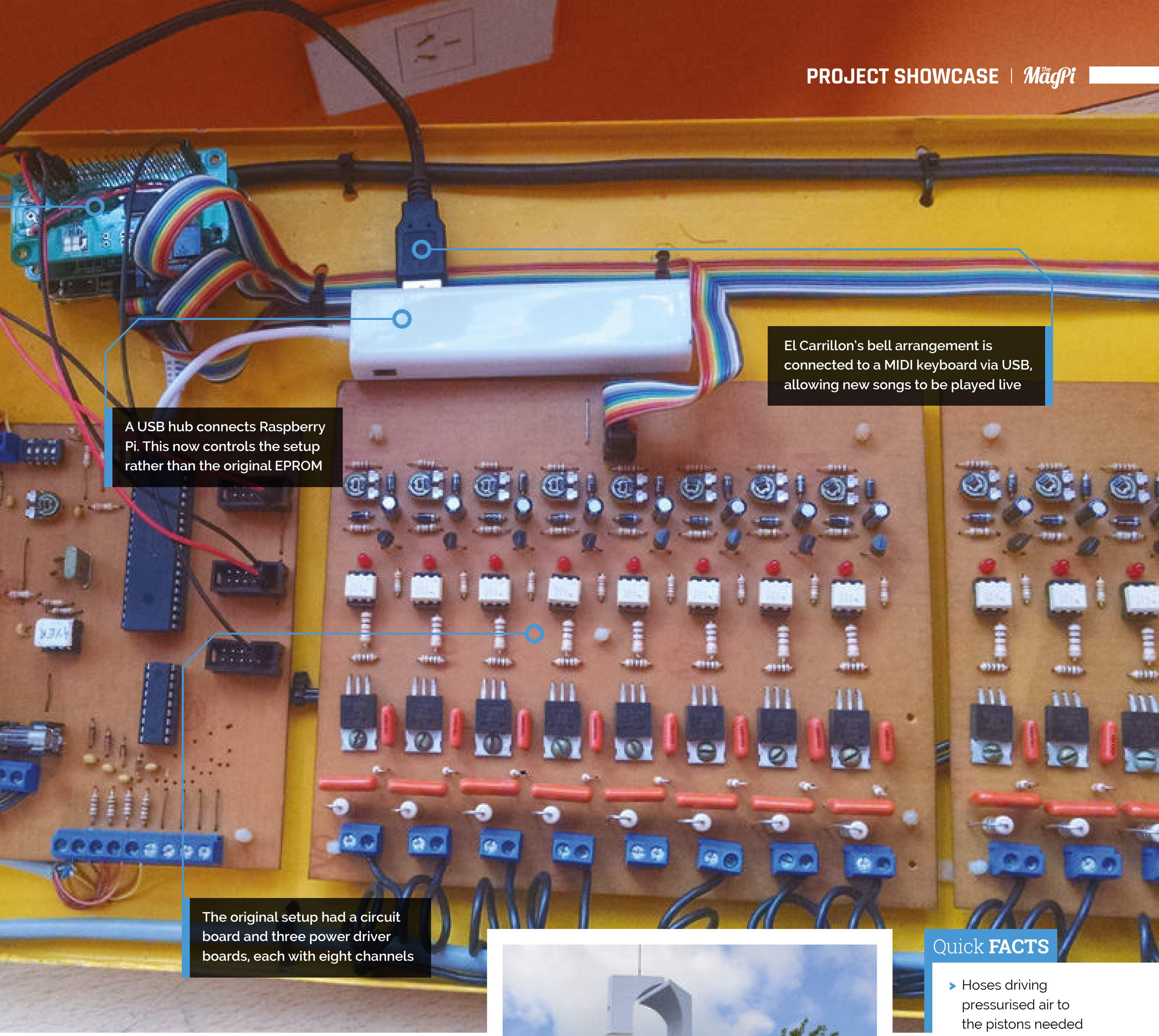
Many years ago, festival organisers created custom hardware with a PIC (programmable interface) microcontroller to control 18 tuned bells. Each bell is associated with a musical note, from A3 to D5 with all the semitones. Until its long overdue update, the tower's 18 bells had rung the tune to *Ayer*, also known as *Yesterday* by The Beatles. They now have a brand-new repertoire of MIDI-based tunes, including the theme from *Star Wars*.

▲ The El Carrillon bell tower forms a striking backdrop to the flower festival and other cultural events

For Gerardo Richarte, the originator of the project, there was a little extra pressure: his dad is on the board of the NGO that organises Fiesta Nacional de la Flor, and challenged his son to come up with a way to update the bells so different songs could be played.

Ringing the changes

With the challenge accepted, Mariano Martinez Peck explains, "We chose Raspberry Pi because it was inexpensive, yet powerful enough to run Linux, Python, and VA Smalltalk. We could find ready-made HATs that actually matched the pinout



El Carrillon's bell arrangement is connected to a MIDI keyboard via USB, allowing new songs to be played live

A USB hub connects Raspberry Pi. This now controls the setup rather than the original EPROM

The original setup had a circuit board and three power driver boards, each with eight channels

"A website that allowed control, configuration, and playback of MIDI songs on the bell tower"

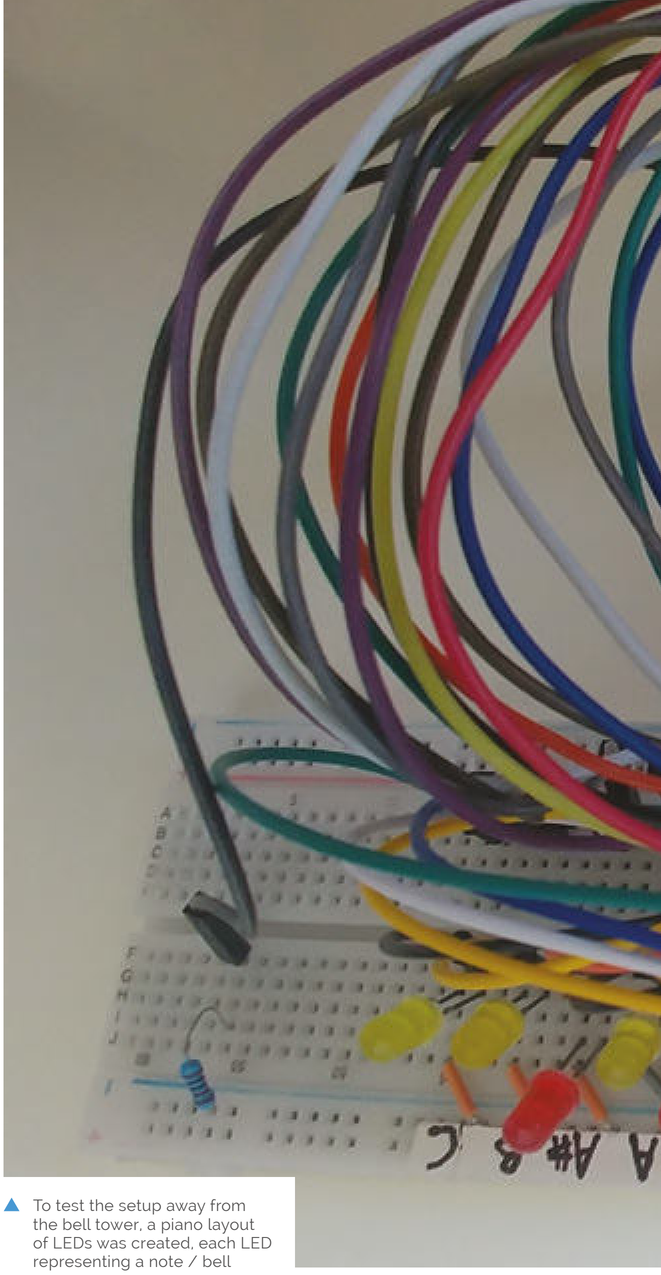
of the existing flat cables without much hacking, and only a minimal amount of other hardware was needed. In addition, there was plenty of documentation, materials, tutorials, and GPIO libraries available."

The project aim was to be able to run a mobile-friendly website within Raspberry Pi Zero that allowed control, configuration, and playback of MIDI songs on the bell tower. "In addition,

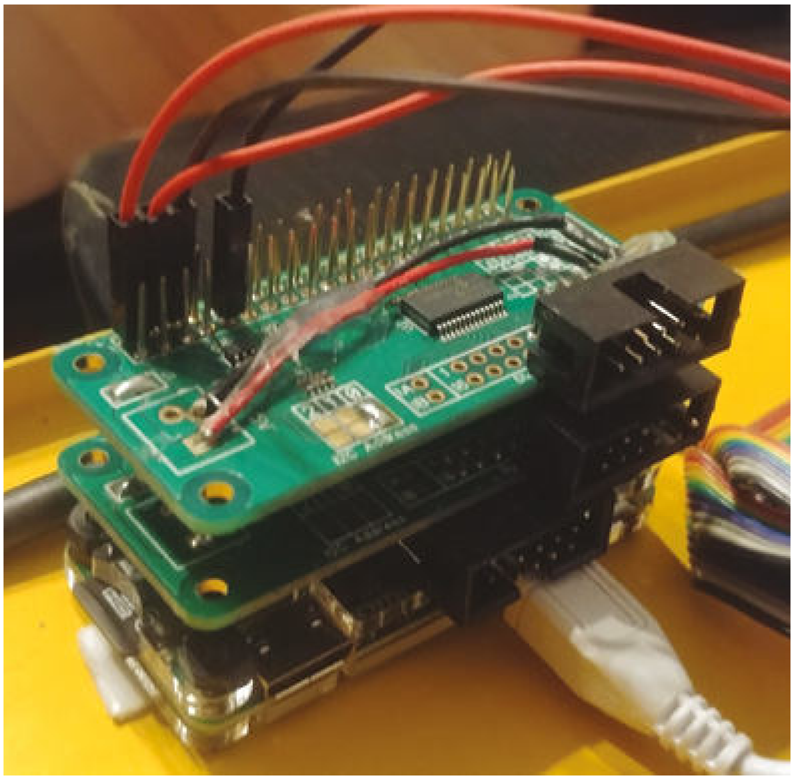
49° Fiesta Nacional de la Flor
Una explosión de naturaleza y color
ESCOBAR CAPITAL NACIONAL DE LA FLOR

Quick FACTS

- Hoses driving pressurised air to the pistons needed overhauling
- Identically mapping the bells using an 'LED piano' helped with testing
- They used FluidSynth MIDI software to check the tunes
- The attached keyboard can record the chimes performances
- Having a fixed deadline drove the makers on



▲ To test the setup away from the bell tower, a piano layout of LEDs was created, each LED representing a note / bell


▲ Two IO Pi Zero boards are stacked on top of a Raspberry Pi Zero to add extra connections

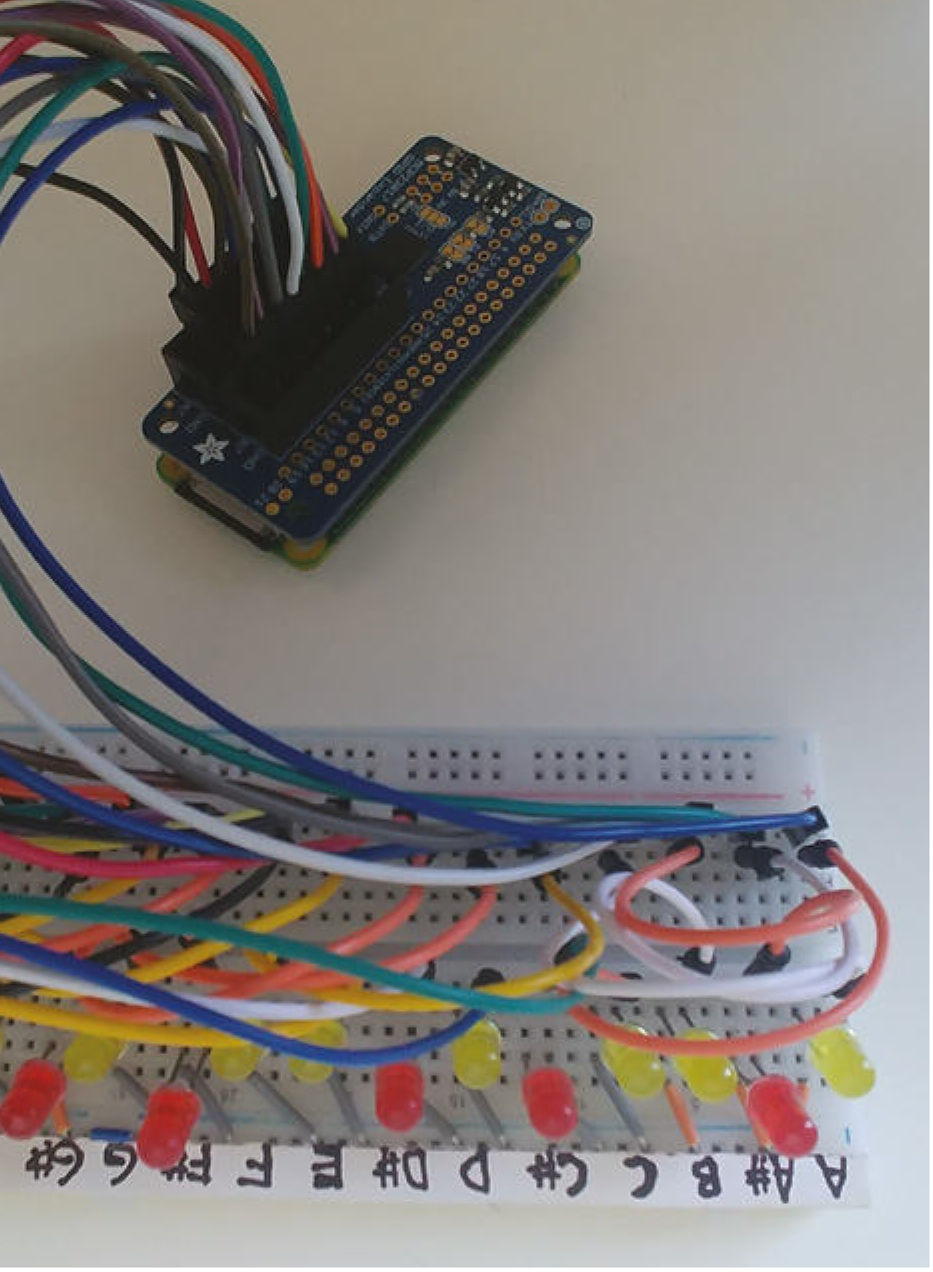
▲ A MIDI keyboard is connected to the setup, enabling a user to play live and record songs

we wanted to allow live playing from a MIDI keyboard," says Mariano. The project developed as a live test and iteration update, but the final build only came together when Mariano and Gerardo's moment in the spotlight arrived and El Carrillon rang out the first new tunes.

Coding a classic

The decades-old chimes were controlled by assembly code. This was superseded by Python when the team made the switch to Raspberry Pi Zero. Mariano explains, "Raspberry Pi allowed us to use Python to directly interface with both the old and new hardware and get the initial project working."

However, the Python code was itself replaced by object-oriented VA Smalltalk code – an environment both Mariano and Gerardo are adept at using. Mariano says, "Smalltalk's live programming environment works really well for fast, iterative development and makes software updates quick and easy without the need for



“El Carrillon’s bells can now play any MIDI file on Raspberry Pi, and the notes of the song will be mapped to the tuned bells”

recompilation that lower-level languages [such as assembly or C/C++] would need.”

El Carrillon’s bells can now play any MIDI file on Raspberry Pi, and the notes of the song will be mapped to the tuned bells. However, as the testing process revealed, some songs are more recognisable than others when reproduced on chimes.

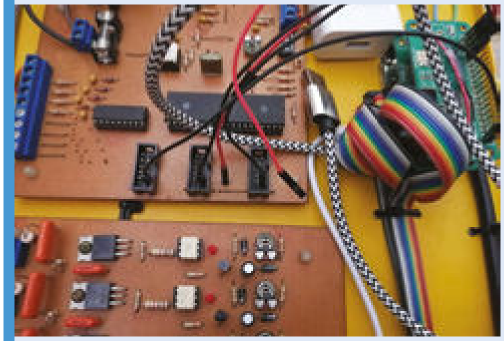
A final feature enabled Gerardo to bag some brownie points with his father-in-law. He recently added a web interface for controlling, configuring, and playing songs, meaning the bells can now be controlled remotely and the song selected via a smartphone app.

Warning!
Heavy objects

Church bells can weigh more than a tonne and are maintained by specialists.

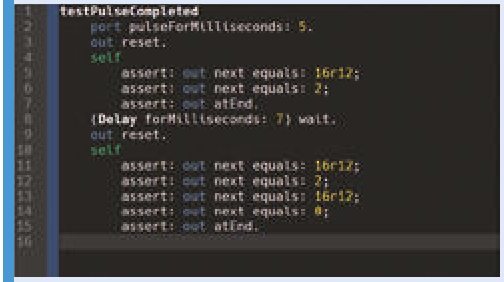
magpi.cc/churchbells

Make your own chimes



01 Originally, the bell tower code was written in assembly language and there was a chip on the board labelled ‘Ayer’ (Spanish for ‘Yesterday’) that held a rendition of The Beatles song. Unfortunately, this was the only song available for playback.

02 A Raspberry Pi Zero, two IO Pi Zero expansion boards (each with an MCP23017 GPIO expander), and the bells’ pre-existing driver module were connected together. Luckily, the power driver board and the flat cables could remain untouched and be reused. The pinout could be easily adapted to the IO Pi Zero boards by just soldering right-angled headers to them.



```
testPulseCompleted
    port pulseForMilliseconds: 5.
    out reset.
    self
        assert: out next equals: 16r12;
        assert: out next equals: 2;
        assert: out atEnd.
    (Delay forMilliseconds: 7) wait.
    out reset.
    self
        assert: out next equals: 16r12;
        assert: out next equals: 2;
        assert: out next equals: 16r12;
        assert: out next equals: 0;
        assert: out atEnd.
```

03 Now with the new hardware and software, you can play any MIDI file on Raspberry Pi and the notes of the song will be mapped to the tuned bells. The code to control the chimes (along with more detailed instructions) can be found **magpi.cc/carrillon**.

Bellagio Water Show

This miniature fountain features water jets synchronised to music. **Phil King** dives in

MAKER

Nick Rogness

Nick works in the telecom industry, but his true passion is spending time with his family and taking on challenging projects as an avid maker.

magpi.cc/bellagio

When his employer wanted a Las Vegas-themed float for the local parade, Nick Rogness was approached by the team for engineering assistance. "The decision to recreate the Bellagio water show started out as many good ideas start… a joke during a brainstorming session!" he recalls. "I knew it was technically possible in principle but would require overcoming some challenging engineering hurdles."

Nick had just 30 days to come up with a technical solution to recreate the water show. "In reality, it ended up taking two weekends of assembling the hardware and two weekends of writing the software," he says.

Pump it up

A pump pushes water from a reservoir (children's paddling pool) through PVC piping attached to water solenoids connected to sprinkler tubing pointed up in the air. A Raspberry Pi controls the solenoids, creating the effect of water jetting out in sync with the music being played.

"A total of eight solenoids were connected back to a mechanical relay, which in turn was controlled by Raspberry Pi," says Nick. Seven out of the eight solenoids were connected to brass reducers to fit into garden sprinkler tubing. The eighth solenoid was a pressure control (relief) valve, which was used to control back pressure in the system.

"When I wanted to 'fire' one of the seven solenoids to shoot water, Raspberry Pi would close the pressure solenoid," explains Nick. This built up pressure in the PVC pipe, at which time Raspberry Pi would trigger a relay to open the desired solenoid so a jet of water would shoot out. "This was required to get any distance with very little water. I also didn't want to burn out the pump, so the relief valve was open when no other solenoid was open."

Water music

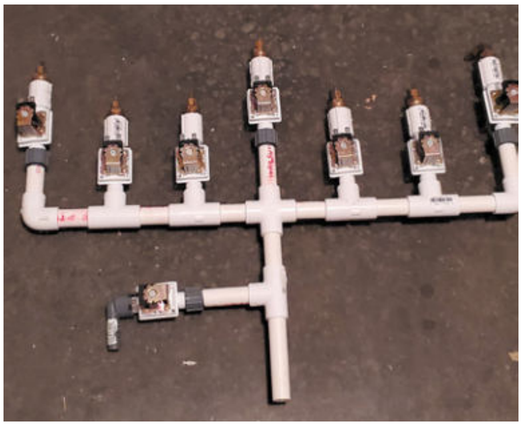
The music is synchronised to the solenoid firing by using FFT (fast Fourier transform) analysis performed on the audio in real-time. "I wrote a sequencer in Python to perform the analysis and determine which solenoids to turn on and off, based on a config file which maps high fidelity signals (bass, mid-range, etc.) to particular solenoids or solenoid groups," says Nick. "In summary, you just put WAV files in a songs directory and start the Python code, which did all the heavy lifting in real-time."

One technical challenge was solving the timing discrepancy between the solenoid firing water and the musical note being heard by the audience. "The water had to be shot out of the jets approximately 600 ms ahead of the audio for the water to appear to be in sync with the music."

Another issue was safety, as mixing water and electricity can be hazardous. "The power for the system was a 12 V automotive battery," reveals Nick, "so I used fuses to protect things, just as you would find in a family vehicle. I also tried to keep the dangerous gear out of reach of the general public."

Everything went well on the day, albeit with a few bugs: "There were certain sequences of musical notes where the FFT analysis would produce changes too rapidly for the back pressure and corresponding solenoid firing to produce much of a water jetting effect." The result was a variance in water height from song to song.

"I rode on the float during the parade, so the public reaction was the most rewarding part of the project for me," he adds. "After people figured out what they were looking at, the responses ranged from laughter to astonishment. The public response made my day and all the efforts of the team worthwhile!"



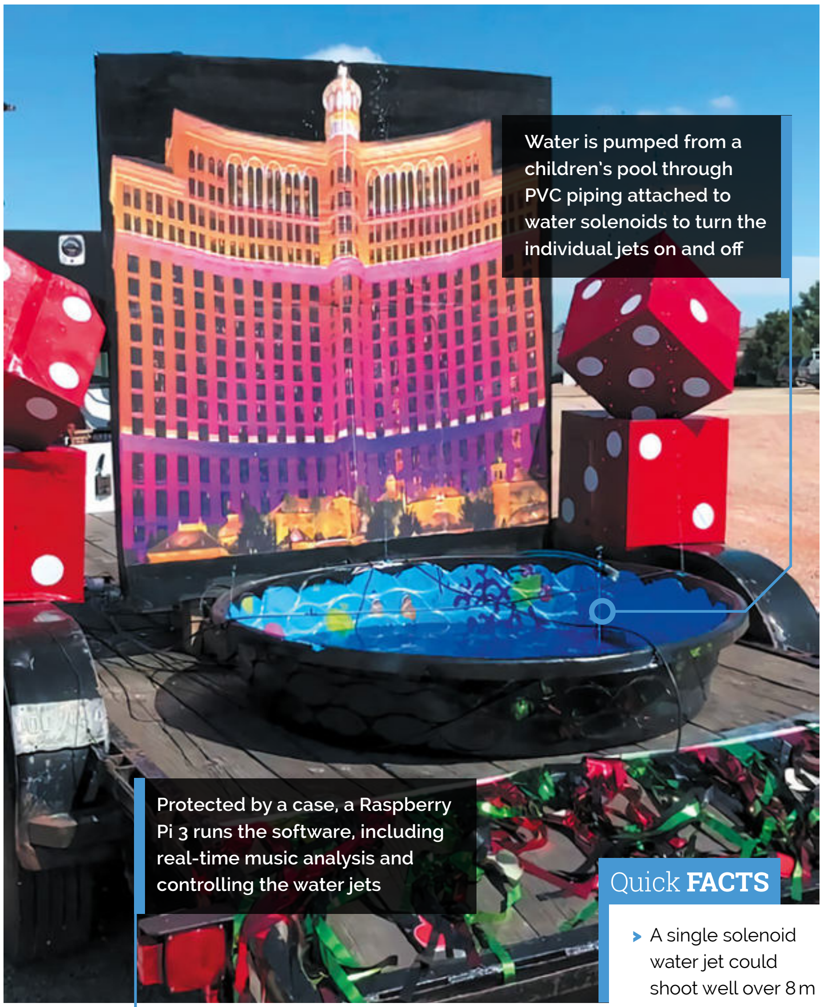
▲ Attached to half-inch PVC piping, each 12V DC water solenoid is connected back to the control board by way of a mechanical relay, operated by Raspberry Pi

Warning!
Electricity & water

Take extra care when combining electricity and water in a project: the two should be kept well apart!

magpi.cc/electricalsafety

A side view of the finished project on the back of the float, ready to join the parade

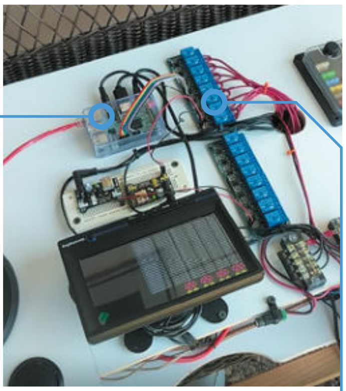

Water is pumped from a children's pool through PVC piping attached to water solenoids to turn the individual jets on and off

Protected by a case, a Raspberry Pi 3 runs the software, including real-time music analysis and controlling the water jets

> Eight solenoids were connected back to a mechanical relay, which in turn was controlled by Raspberry Pi

Quick FACTS

- A single solenoid water jet could shoot well over 8 m
- The water used is in a closed system
- Software is available on GitHub: **magpi.cc/ghbws**
- He'd never worked with water in an electronics project
- Future projects include a security drone and snow-removing robot


Nine relays are used: one to control the water pump, and the others to open and close the solenoid valves

A waterproof fuse box is used to safely distribute power from a 12 V car battery

Raspberry Pine

Think all computers are boring black boxes? You've not seen Raspberry Pine, the desktop computer with a bit of class. **PJ Evans** gets all fancy

MAKER

Neil Shepherd

Neil has been working with computers since the 1970s. Upon retiring, he joining U3A (University of the Third Age). Their electronics group rekindled his interest in Raspberry Pi.

To many, a broken laptop would be a source of annoyance and hassle, but not Neil Shepherd, who saw nothing but opportunity. After a brief respite at the bottom of his wardrobe, Neil dismantled the laptop looking for useful spares and maybe some inspiration. He hit upon the idea of building a desktop computer that would fit in with the surrounding furniture, rather than being yet another drab black box. "What do you do with a 17-inch LCD panel? Put a frame round it," he says. "It's got to be different, so wooden… pine to be precise. Add a Raspberry Pi 3B+ and my love of awful puns gives us Raspberry Pine."

Wood-n't it be nice

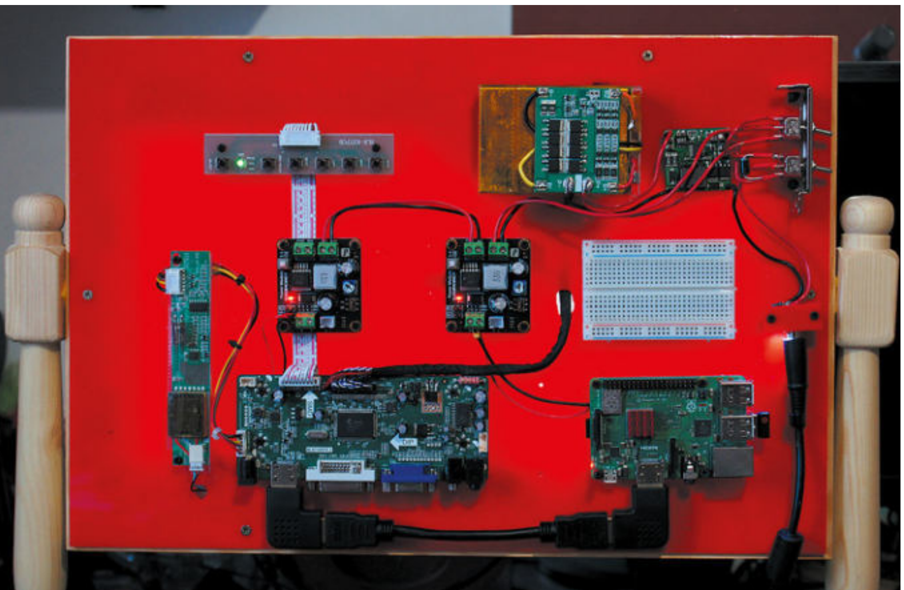
The screen is mounted in a custom pine frame and stand. The frame is built using strips of small pine architrave backed by pine strips, offset to give a suitable rebate for the panel to fit and hiding the border. The side supports are modified pine staircase spindles. A Perspex sheet is used to mount the electronics and, when powered, the panel backlight creates a warm glow. Finally, short lengths of foam draft excluder secure the panel sandwiched in the frame.

The LCD panel has a driver board with HDMI, DVI, and VGA inputs. This is connected to the project's Raspberry Pi 3B+ by a short HDMI lead, with left- and right-handed adapters. This caused a headache for Neil: "I think the biggest challenge was sourcing the LCD panel driver board from a UK supplier on eBay. Unfortunately, the only documentation you get is from the tiny screen printing on the board itself."

The rest of the electronics supply and manage power. The main supply is the original PSU brick from the donor laptop, providing 19.5 V at 4.5 A. Two DFRobot PSU boards are used to drop the 19.5 V to 5 V for Raspberry Pi, and 12 V for the LCD driver board, enough to power the backlight.

There is also the option of battery power, from three Li-ion cells recovered from another laptop. A battery management system (BMS) board sits on top of it and ensures the cell charge states are balanced. Finally, a charger board provides the correct constant current and voltage charge profile.

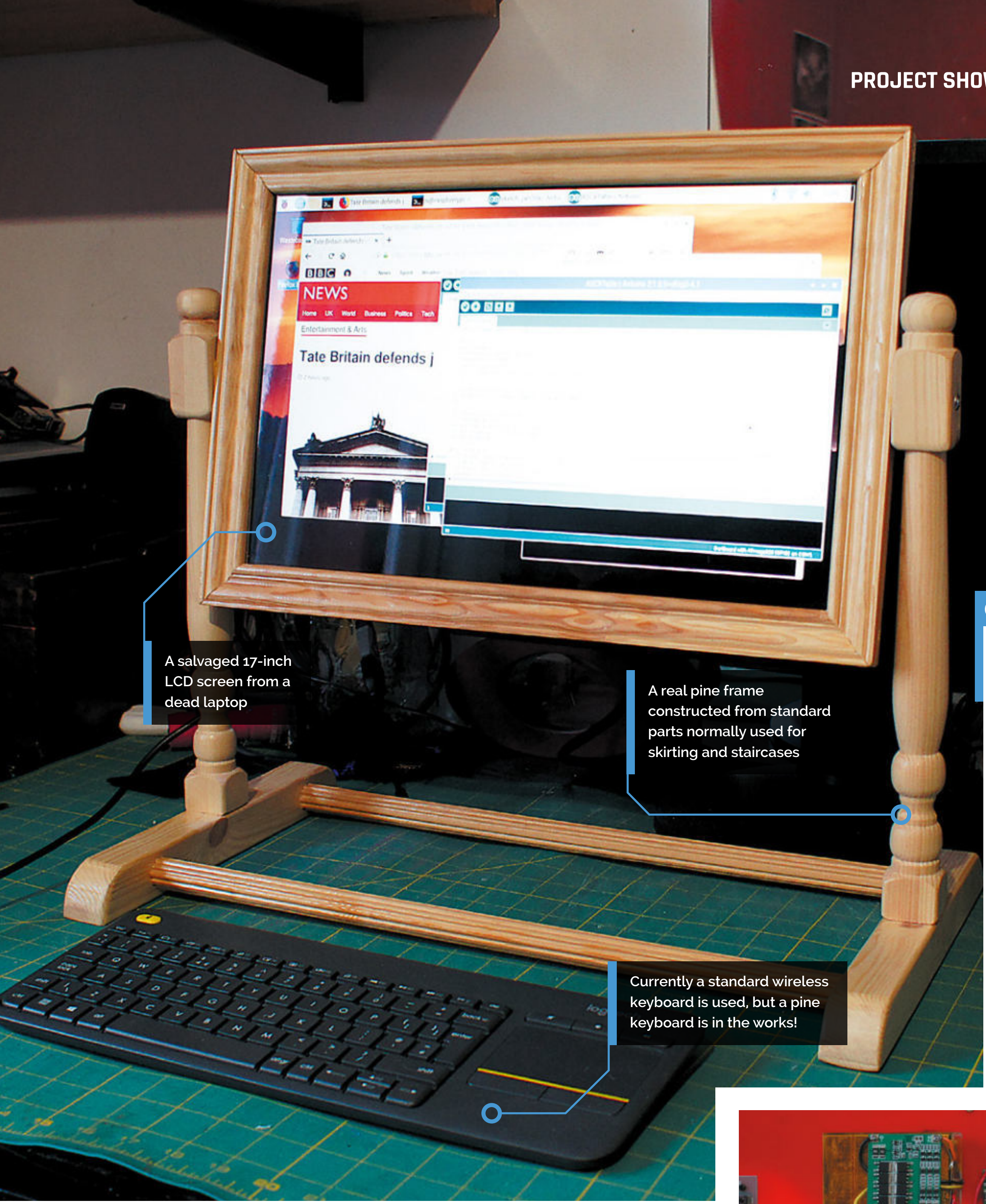
" What do you do with a 17-inch LCD panel? Put a frame round it. It's got to be different, so wooden… pine to be precise "


The rear is an exposed collection of controller board and, of course, the project's Raspberry Pi 3B+

A tree-mendous result

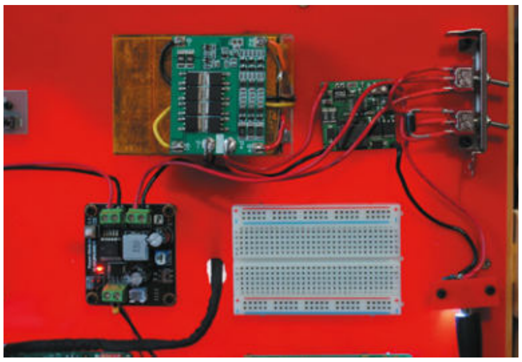
The result is a very different take on the classic desktop project: a computer that would look just as at home on the bedroom dressing table as it would in the study. Neil was certainly pleased with the results, as well as an unplanned effect: "The choice of raspberry-coloured Perspex as the support for the electronics produced a pleasant surprise. When powered on, the LCD backlight gives a raspberry glow."

Regarding planned further refinements, he tells us: "When the Raspberry Pi 4B+ gets the update to boot from USB, there may be a rebuild to refine the layout so the SSD doesn't stick out the side. Currently, the donor laptop keyboard is being developed with an Arduino Due and more pine. There are thoughts of matching mouse and speakers."

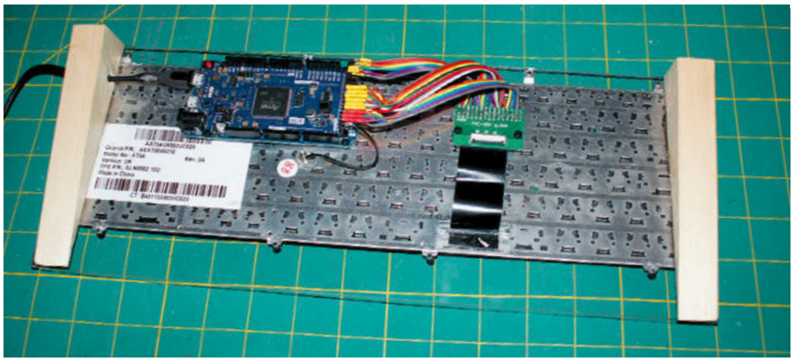

A salvaged 17-inch LCD screen from a dead laptop
A real pine frame constructed from standard parts normally used for skirting and staircases
Currently a standard wireless keyboard is used, but a pine keyboard is in the works!

Quick FACTS

- The screen is a 2007 HP 17 laptop panel
- The computer will soon be joined by a pine-encased keyboard
- You can run the computer from the mains or battery
- The pine used is off-the-shelf, with small modifications
- Neil's next project is to record the animal that keeps pooing in his garden!



Neil has thoughtfully added a breadboard and control switches for ease of use



A matching keyboard, also from the laptop, is a work-in-progress

SUBSCRIBE TODAY FROM ONLY £5

SAVE UP TO 67%

USA OFFER!
12-ISSUES NOW $60*

OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
Issue 91 | March 2020 | magpi.cc
Raspberry Pi 4 2GB upgrade!
Starter Electronics
#MonthOfMaking
Let's build together!
Top 10 Wearable Projects
40 PAGES OF PROJECTS & TUTORIALS

Subscriber Benefits

- **FREE Delivery**
  Get it fast and for FREE
- **Exclusive Offers**
  Great gifts, offers, and discounts
- **Great Savings**
  Save up to 67% compared to stores

Rolling Monthly Subscription

- Low monthly cost (from £5)
- Cancel at any time
- Free delivery to your door
- Available worldwide

Subscribe for 12 Months

£55 (UK)	$60 (USA)
£80 (EU)	£90 (Rest of World)

Free Raspberry Pi Zero W Kit with 12 Month upfront subscription only (no Raspberry Pi Zero W Kit with Rolling Monthly Subscription)
*Subject to change or withdrawal at any time

USA offer: **magpi.cc/usa**

Subscribe by phone: **01293 312193**

Subscribe online: **magpi.cc/subscribe**

Email: **magpi@subscriptionhelpline.co.uk**

JOIN FOR 12 MONTHS AND GET A

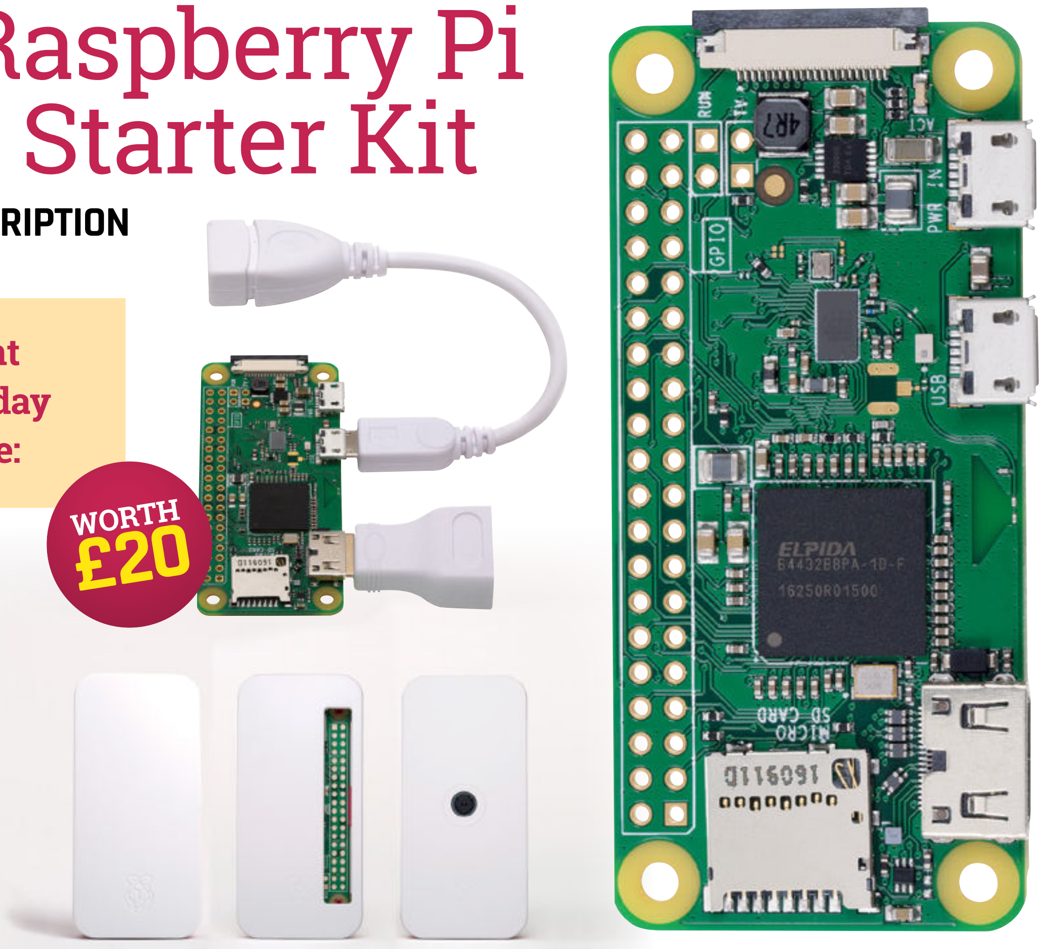
FREE Raspberry Pi Zero W Starter Kit

WITH YOUR SUBSCRIPTION

Subscribe in print for 12 months today and you'll receive:

- Raspberry Pi Zero W
- Raspberry Pi Zero W case with three covers
- USB and HDMI converter cables
- Camera Module connector

Offer subject to change or withdrawal at any time



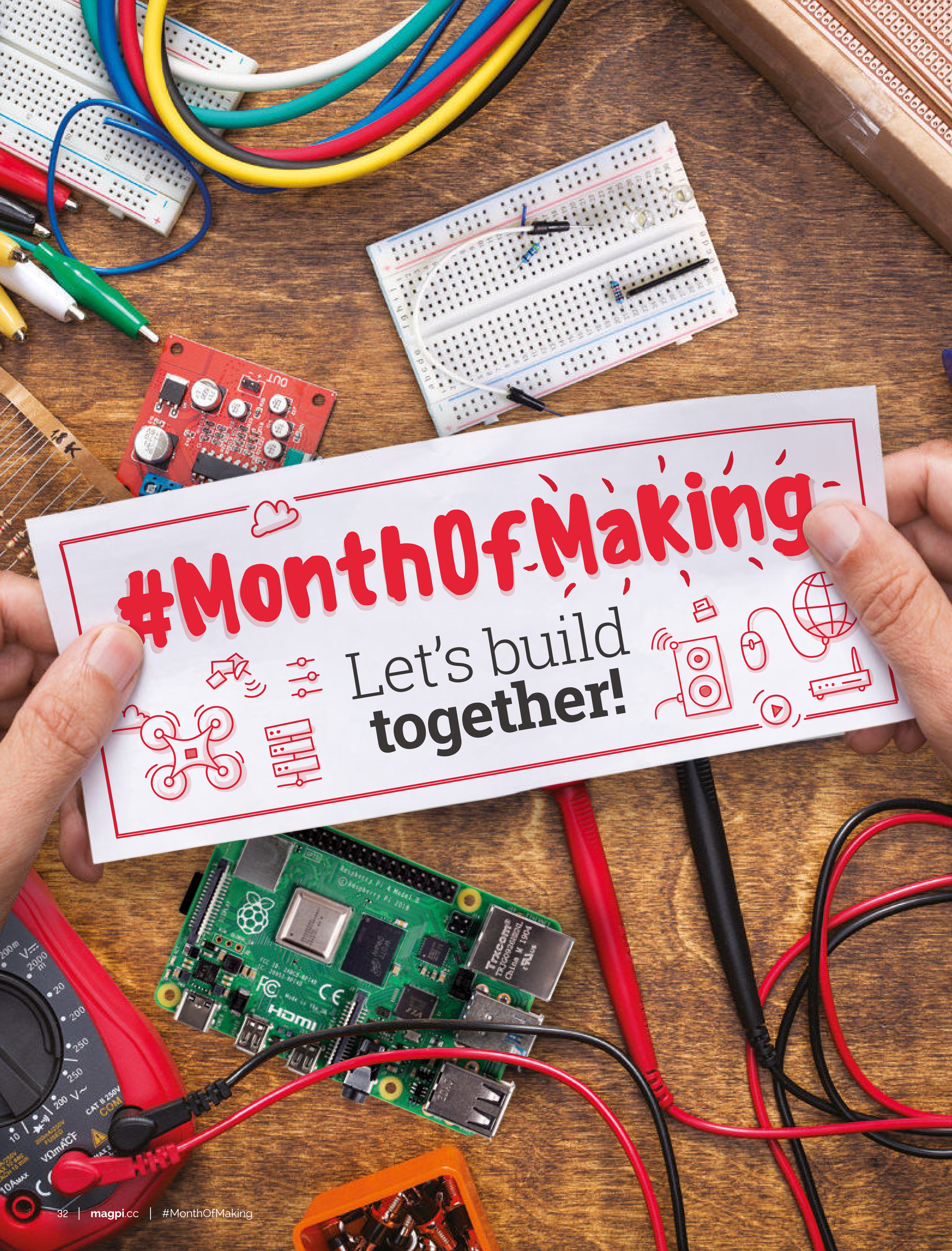
Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

From £2.29

Available on the App Store

GET IT ON Google Play


#MonthOfMaking
Let's build
together!

We're building and sharing again in the #MonthOfMaking. Let **Rob Zwetsloot** show you how

If you're reading this magazine, it's safe to say you like making in some way. The hobby has exploded in popularity over the last few years, thanks in no small part to a burgeoning online community and the introduction of low-price computing with Raspberry Pi.

Last year we decided to celebrate making with a month-long online event called #MonthOfMaking. The idea was simply to get people to share what they're making online, whatever it was. Whether you're turning on your first LED with code or sending rockets to the moon, we want to create a space where you can share your proud achievements. So, let's get making.

What is #MonthOfMaking?

The #MonthOfMaking is simply an excuse to get people inspired to make something. And by make, we mean electronics, engineering, arts, and craft projects. Get your creative powers buzzing and make something that you can show to the world.

There's no level of skill threshold to participate either. We like to think if you've been wanting to start to learn, this can be your jumping-on point. By sharing your builds with the community, you can learn and grow. Here are some simple rules to sum it all up:

1. Find a new project, continue with one you're working on, or finally crack on with something you've been putting off.
2. Take pictures of your build progress and share it online with the hashtag #MonthOfMaking.
3. If you can help someone with a problem, give them a hand.
4. Have fun!

Getting ideas and inspiration

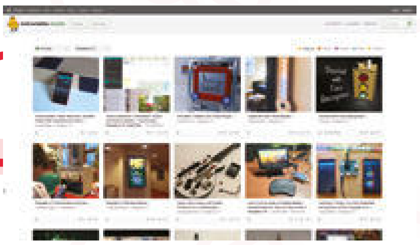
What do you want to build?

We've all been there. Sat down at a work bench or desk, staring at some components and thinking… what can I make with this? What would I like to make? Like any other creative pursuit, you'll need some inspiration. If the projects in the magazine haven't inspired you, then here are some website suggestions…

Tools for the job

Not everyone has a full workshop of tools and gizmos, especially if this is your first foray into making. Check out issue 89 of *The MagPi* (**magpi.cc/89**) for our 50 Tips & Tools feature, which has plenty of tool suggestions.

Project websites



▲ Instructables

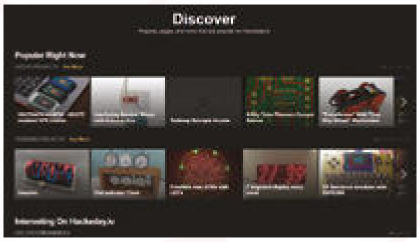
Instructables is one of the oldest sites out there for finding amazing project guides and ideas, and we've been fans of it for years. The best part is you can search by specific project types as well, including Raspberry Pi if you'd like to keep it on-brand. They've recently added more arts and crafts stuff if you fancy trying your hand at knitting.

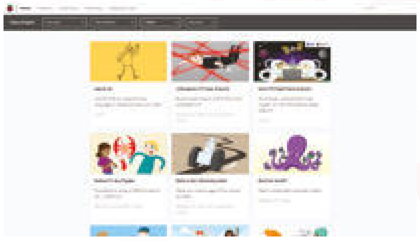
instructables.com

▼ Hackaday

For more serious hacks for more advanced makers, Hackaday has some great projects that really take a deep dive into a project. If you're curious as to the limits of electronics and programming, this may be the place to look. Equally, if you want to do something huge with a lot of computer power, this should be your first stop.

hackaday.io





▲ Raspberry Pi Projects

There are so many amazing things on the Raspberry Pi Projects site that can help you with your first steps in just about any field of making. It's also home to loads of great and simple home-grown projects that are perfect for young makers and older makers alike.

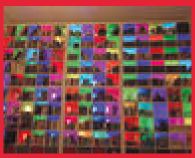
rpf.io/projects

Project ideas

Magic Mirror

SKILLS: Carpentry, programming

magpi.cc/90



NeoPixel display lights

SKILLS: Programming, circuits, soldering

magpi.cc/neopixelpi

Planning your build

Before getting hands-on with a project, do your preparation

01 Read and understand

Basing your build on a tutorial you've seen? Seen a few things you'd like to combine into something else? Always make sure to read the instructions you've found properly so that you know if it's within your skill level.

02 Order supplies

Write a list of what you need. Always double-check you have the component you think you have. Sometimes you may need to buy from separate places, so just make sure the delivery times work for you.

03 Follow along and be safe

Need adult supervision for a project? Absolutely get some. Even adults need to be wary, so always take safety precautions and wear protective clothing when needed. Make sure to follow any tutorials you've found as closely as you can.

Tip!

It's fine to drop a project if you're not enjoying it. Learn, and move on.

Online stores

Pimoroni

Paul from Pimoroni was the one who created the logo for Raspberry Pi, and the Pimoroni online store is one of the oldest Raspberry Pi-related maker stores. As well as the famous Pibow case made from laser-cut acrylic, they've created a huge number of fun and amazing Raspberry Pi and maker kits and accessories over the years, and stock a lot of Adafruit's stuff for UK buyers as well.

shop.pimoroni.com

Adafruit

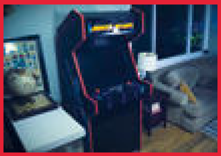
A legendary stop for North American makers, not just for all their excellent products, but also for an amazing website full of fun tutorials to give you further inspiration. A lot of their tutorials even have links to product bundles so you can start building them quickly.

adafruit.com

The Pi Hut

Another great UK store full of great kits and Raspberry Pi- and maker-related products. You can also find official Raspberry Pi swag there, like shirts you could otherwise find in the Cambridge Raspberry Pi store.

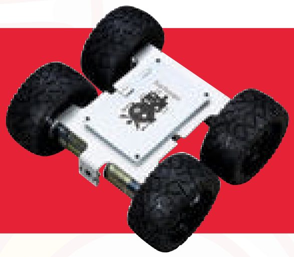
thepihut.com



Arcade machine

SKILLS: Carpentry, CAD, designing, programming

magpi.cc/63



Robot kit

SKILLS: Construction, robotics, programming

magpi.cc/monsterborg

Becoming part of the community

Makers, Raspberry Pi users, coders, developers, and more can easily be found online!

Being part of a community is an amazing thing. Getting to make new friends, learn new skills, and be inspired by those around you is wonderful. Here are some of the places you can find Raspberry Pi fans and makers, or get a bit of help if needed.

Offline communities

Not really into being online? Raspberry Jams, Code Clubs, and CoderDojos, along with other Raspberry Pi and maker events, can be found around the world! Check out our events calendar on page 88 for more details on upcoming events!

Online community

Raspberry Pi
@Raspberry_Pi
The official Twitter account for the Raspberry Pi Foundation. News and info about our low-cost mini PC.
Cambridge, UK raspberrypi.org Joined May 2011

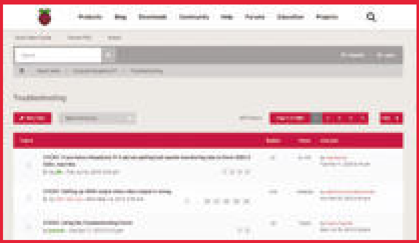
▲ Twitter

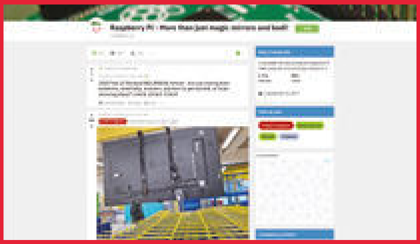
Here at *The MagPi* and at Raspberry Pi, we love to see and share stuff on Twitter. It's a great way to get a lot of people to interact with your builds, and also get some help if you need it. Along with using the #MonthOfMaking tag, you can keep an eye out for #MagPiMonday posts, and don't forget to tag anything you build with #MyLatestBuild.

▼ Raspberry Pi forums

The official forums are an incredible place to go for finding out more info about Raspberry Pi, or even getting some help. There's always a buzz of activity and a lot of the users are very friendly. Make use of the forum search function and you may find the answer to just about any tech issue.

rpf.io/forums





▲ Reddit

The Raspberry Pi subreddit is a good way to keep up with all things Raspberry Pi at your own pace, as well as seeing incredible builds from all around the world. You can even post your own stuff on here – just make sure to use the recommended tags so it hits the right audience!

magpi.cc/reddit

Project ideas

Laptop

SKILLS: Construction, 3D printing, customisation

magpi.cc/14

Video doorbell

SKILLS: Upcycling, programming, networking

magpi.cc/doorbell

Getting help

If you need help, there's plenty available from other makers online

Even the greatest makers need help sometimes – their secret is that they have found great ways to get support when they're stuck. Aside from having the right friends and the right reputation, there's no real secret as to where you can get your help from on the internet.

Search engines

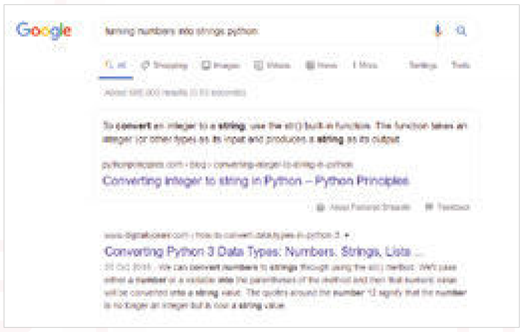
Your first port of call should always be your preferred search engine. Whether you want to know if something is possible (and maybe get a tutorial for it) or need help with a specific coding issue, you're likely to find a solution.

Be careful with your keywords, though – being specific can help you out a lot, and if you know the exact wording or phrasing of the issue you're searching for, it can make sorting out your problems much quicker.

Social media

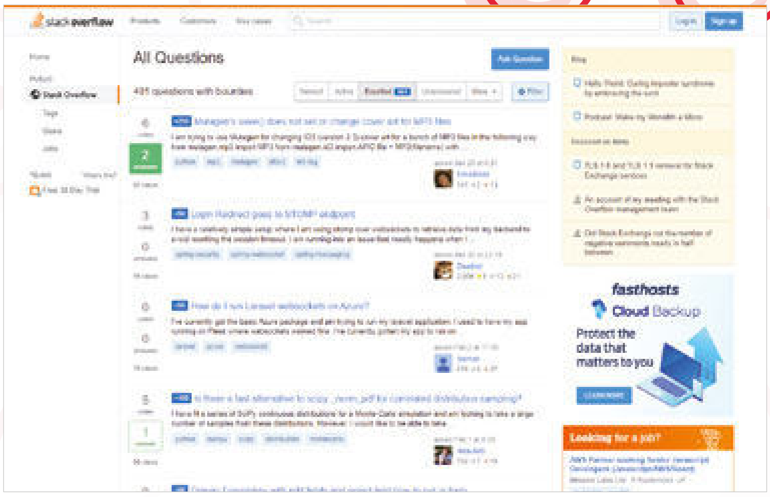
While some people may refer to this as the 'lazy web', we think it's totally fine to ask a question on social media while you try to look for a solution elsewhere. Sometimes people can parse your language and ask the right questions to get to the bottom of a problem faster than trying to refine your Google search.

If there's a relevant hashtag, make sure to use it. During March, if you use #MonthOfMaking, we'll try to elevate the issue and get experts to help you out.


turning numbers into strings python

Converting integer to string in Python – Python Principles

Converting Python 3 Data Types: Numbers, Strings, Lists ...


All Questions

481 questions with bounties

▲ Stack Overflow gives people incentives to answer questions

Question sites

There are a few online sites where you can ask questions. For Raspberry Pi-related queries, you can always visit the Raspberry Pi forums that we mentioned before. However, for all code-related issues, you can always check out the amazing Stack Overflow. It's used by professionals and amateurs alike to answer their questions.

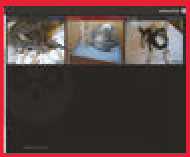
For physical making stuff, it depends on what you're building. The Replica Prop Forum is a good place to start, and cosplayers will have some experience with wearables as well. Search engines are your friend in finding what you need here.

◀ Learning Google-Fu, specifically for phrasing tech questions, is a valuable skill

PiGrrl 2 - handheld retro gaming

SKILLS: 3D printing, construction, programming

magpi.cc/pigrrl2



CCTV

SKILLS: DIY, programming, networking

magpi.cc/motioneyeos

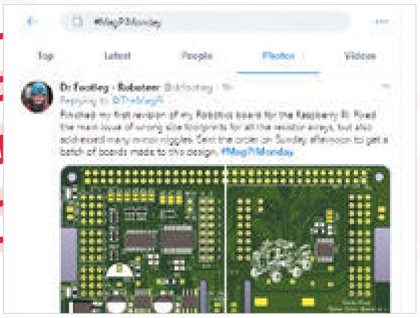
Sharing your projects

The point of #MonthOfMaking is to share what you're building! Here are our tips on how to post your projects online

Send it to us!

If you'd prefer to send us your builds, send us an email to **magpi@raspberrypi.com** with the subject line #MonthOfMaking.

Posting your projects

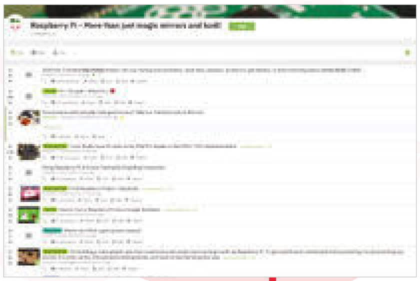


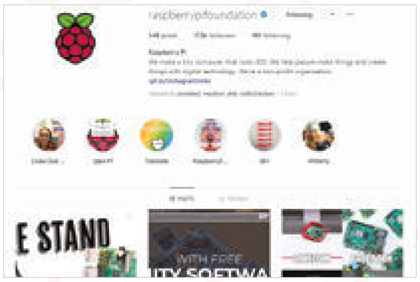
▲ Twitter

Posting on Twitter is the best place to share your project. Include a brief description, pictures, and any relevant links if you have a blog for the build, tagged with #MonthOfMaking and #MyLatestBuild. You can also tag in **@TheMagPi** and we'll make sure to share it ourselves, and maybe even feature it in the magazine!

▼ Reddit

There are many relevant subreddits for different project types – although the Raspberry Pi subreddit is a good place to go for any Raspberry Pi-related projects you want to post. Upload some images to Imgur when you've finished your project, mark your post as 'Show and Tell', and it's ready to show! We like to keep an eye on Reddit, so we'll look out for any #MonthOfMaking posts!





▲ Instagram

We love using Instagram, and posting amazing project builds on there is a great way to get them seen, and to find other amazing folks who are posting build photos. Using the #MonthOfMaking hashtag will get it seen by folks looking at that tag and if you want to post more photos of your build, you can always make it into a story using the story feature!

Project ideas

Media PC

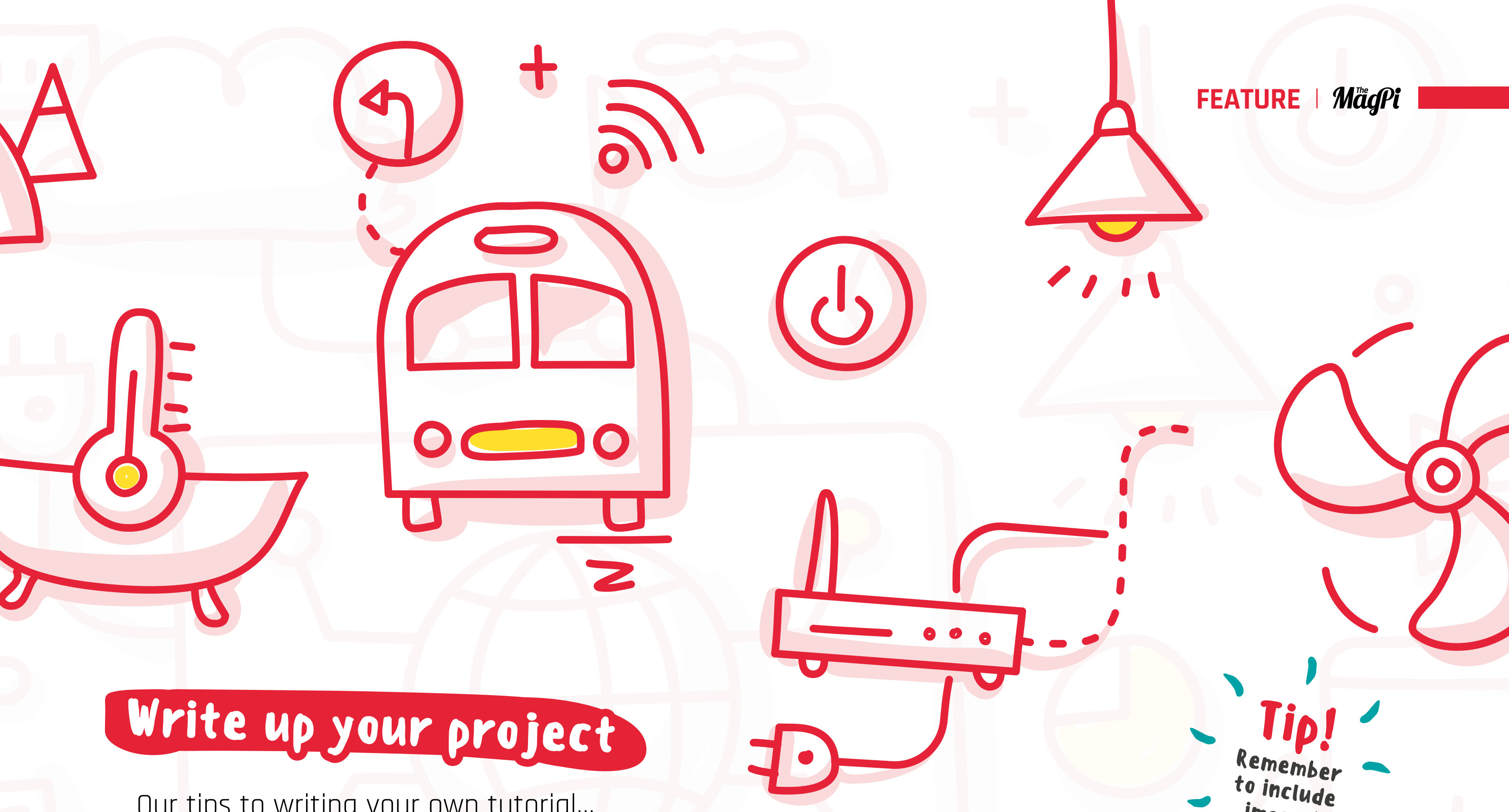
SKILLS: Programming, networking

magpi.cc/81

File server

SKILLS: Networking

magpi.cc/85



Write up your project

Our tips to writing your own tutorial...

Tip! Remember to include images!

01 List of components

We like to first make sure that people know what they need for a project. We'll list specific components like buttons, add-ons, HATs, etc., along with basic components such as wire and resistors, and sometimes add tools you might need as well. This gives people an idea if they have the right stuff to make something.

02 Software instructions

Not every Raspbian build is the same, so make sure you describe how to install any extra software or Python libraries that are needed to complete your project. This includes any Terminal prompts people will have to use, and any software configuration. Not everyone will know exactly how specific bits of software work.

03 Getting any code

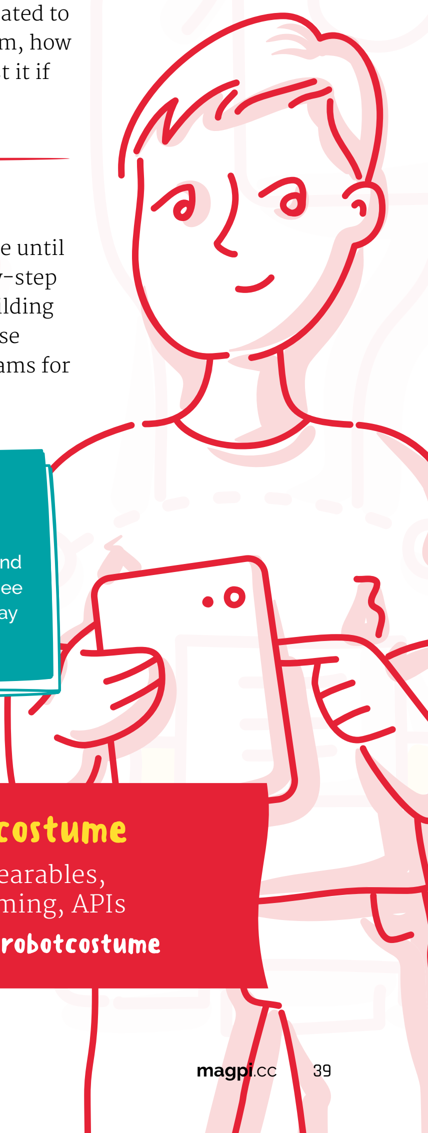
Got code? Put it on GitHub, and include a link to it on your write-up. If it's not huge, you can put it in your tutorial. We suggest putting it at the end usually. Give folks any other instructions related to the code (i.e. where to put it in the file system, how to launch it at boot, etc.) and get them to test it if they can.

04 Building the hardware

We usually leave building the hardware until the end, especially if it's a big build. Step-by-step instructions can be really useful if you're building a full on structure or device, and we like to use Fritzing (**fritzing.org**) to create circuit diagrams for people to follow along to.

Be in The MagPi!

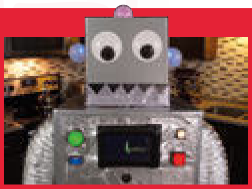
We'll be keeping an eye on all the amazing things you make over the #MonthOfMaking, and we'll be sure to feature all the great stuff we see in the magazine in some way. Your creation may even end up as a project showcase.



Voice assistant

SKILLS: AI, programming

magpi.cc/voiceassistant



Robot costume

SKILLS: Wearables, programming, APIs

magpi.cc/robotcostume

Wireframe presents
Build Your Own
FIRST-PERSON SHOOTER
in Unity
Learn Unity • Create enemies • Design levels • Make Zombie Panic
BUILD YOUR OWN FIRST-PERSON SHOOTER IN UNITY
The theory behind first-person hitboxes
Creating a deployable gun turret in Unity

Build a seismograph with Raspberry Shake

Make an earthquake detector using Raspberry Pi and Shake sensors, then connect to a global network to get involved in citizen science at its finest

MAKER

Lucy Hattersley

Lucy is editor of *The MagPi* magazine and part-time tinkerer. She is quite happy just to hang out and do nerd stuff.

magpi.cc

You'll Need

- Raspberry Pi 2 / 3 Model B (or Zero). See supported models: **magpi.cc/shakespecs**
- Shake RS1D board **magpi.cc/shakers1d**
- Racotech RGI-20DX geophone **magpi.cc/shakesensor**
- Enclosure: laser cutter / 3D printer files **magpi.cc/shakeenclosure**
- Screws, stand-offs, washers **magpi.cc/shakescrews**
- microSD card
- Raspberry Shake OS **magpi.cc/shakeos**

We love Raspberry Shake here at *The MagPi*. This geology project uses a powerful geophone sensor attached to a Raspberry Pi to detect earth tremors.

The geophone converts ground movement into voltage; this analogue signal is then converted into a digital read-out (by the Raspberry Shake board) and the data is stored on Raspberry Pi.

Raspberry Shake is a great project for budding geologists and citizen scientists because it's relatively simple to assemble (although you do need to be careful to handle and level the parts correctly).

Once built, it's low maintenance, sitting in a quiet part of your home or office, waiting for the earth to move. And all that time, Raspberry Shake is gathering data, which you can investigate using the new web interface – or you can dive in and play around with the data directly.

We interviewed Branden Christensen, CEO of Raspberry Shake and seismologist, back in 2018 (**magpi.cc/shakeinterview**). We also did a tutorial in *The MagPi* issue 60 (**magpi.cc/60**).

In the last two years, Raspberry Shake has come along leaps-and-bounds and it now has a powerful web interface, app interface, and a thriving international community. We think it's time to revisit Raspberry Shake.

▲ The Racotech RGI-20DX geophone is the heart of the Raspberry Shake device. It's connected to Raspberry Pi using the Raspberry Shake board

This month we're looking at assembling Raspberry Shake and sharing your data with the wider Shake community.

You can buy all the parts for Raspberry Shake separately (see the 'You'll Need' info) or pick up a turnkey system with all the parts included (**shop.raspberryshake.org**). You can even buy a fully assembled system, but we think that takes all the fun out of things.

01 Wire up the geophone

Start by wiring up the RGI-4.5Hz geophone. Ours has two wires: grey and blue. Make sure the wires are twisted and connect the grey cable from the positive '+' connection on the geophone to the '+' pin on the RS1D Raspberry Shake board. Next, connect the blue wire to the '–' connection. Take care not not over-tighten the screws, otherwise you may damage the wires.

" You can investigate the data gathered using the new web interface "

02 Put Raspberry Pi in the enclosure

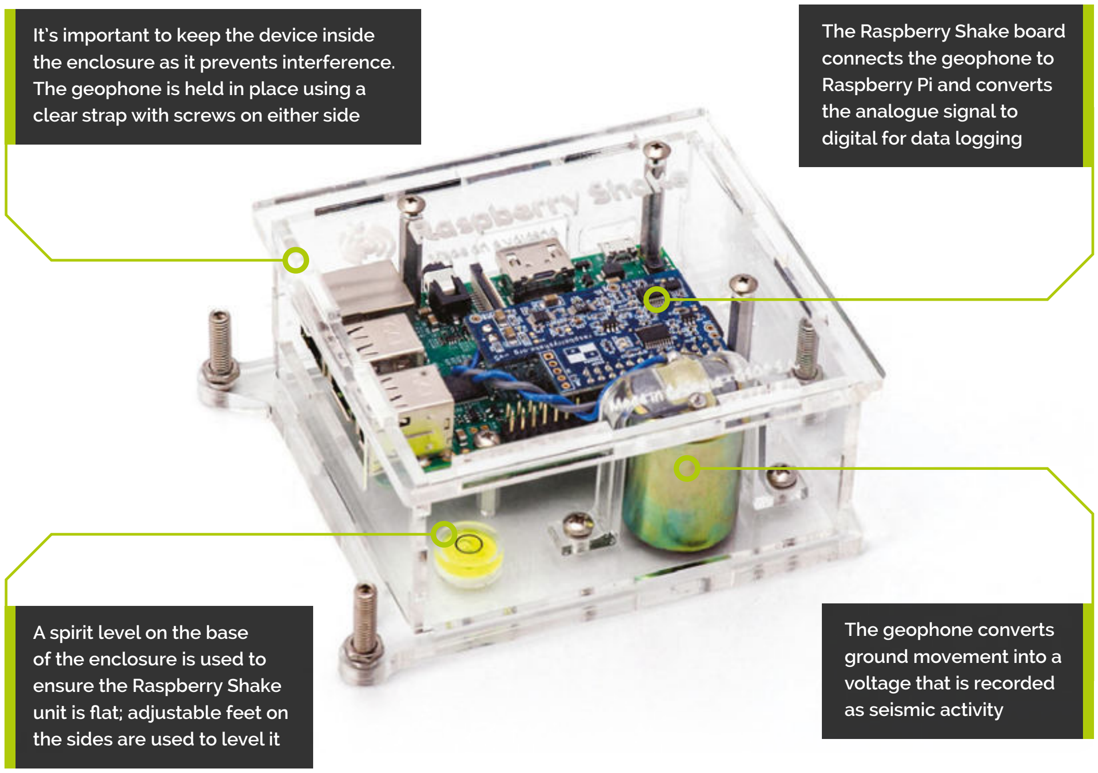
Take the bottom of the enclosure and attach the four shorter stand-offs. Tighten them by hand. Place your Raspberry Pi board on top of the four stand-offs using the holes in the Raspberry Pi.

On top of three of the holes, you need to place a washer and the longer stand-offs (the hole in the middle has just a washer and screw). Take a look at the assembled Raspberry Shake (**Figure 1**) to see which one doesn't need the large stand-off. Now insert your microSD card. If you bought it

Figure 1

It's important to keep the device inside the enclosure as it prevents interference. The geophone is held in place using a clear strap with screws on either side

The Raspberry Shake board connects the geophone to Raspberry Pi and converts the analogue signal to digital for data logging



A spirit level on the base of the enclosure is used to ensure the Raspberry Shake unit is flat; adjustable feet on the sides are used to level it

The geophone converts ground movement into a voltage that is recorded as seismic activity

from Raspberry Shake, it will be pre-installed with Raspberry Shake software. Otherwise, flash a card with the image file (**magpi.cc/shakeos**).

03 Attach the geophone

Place the geophone in the hole on the bottom of the enclosure with the wires on the top. Separate the two wires so there is a gap between them. Now place the plastic strap on top of the geophone to hold it in place. Use two washers and two screws to fix the clear plastic strap to the bottom of the enclosure.

04 Attach Raspberry Shake

Connect the RS1D Raspberry Shake board to Raspberry Pi's GPIO pins. The board has only a 26-pin header (like the original Raspberry Pi Model A and B); most Raspberry Pi boards have 40-pin GPIO, so you'll need to make sure you are connecting the RS1D Raspberry Shake to the correct pins. The board connects to the end of the GPIO pins where the microSD card is (leaving those pins towards the USB sockets free).

Make sure the Raspberry Shake board orientation is correct (the wires to the geophone should be near to the USB ports). If in doubt, take a close look at the main image above (**Figure 1**).

Now clip in the sides of the enclosure. Look carefully at the holes in each side: the small hole is for the microSD card, the medium hole is for the HDMI port and power, and the large hole is for the Ethernet and USB sockets.

The lid of the enclosure has three holes in it, which will line up with the long stand-offs (from Step 02). Use three screws to hold the lid in place. It's recommended to ensure the Raspberry Pi Shake is fully enclosed to prevent any wandering of the results.

05 Levelling Raspberry Shake

The Raspberry Shake enclosure comes with three holes protruding from the sides. These are

Top Tip

DIY tutorials

Raspberry Shake sells a turnkey system, but if you'd rather build everything from scratch, then take a look at its wide range of DIY tutorials. **magpi.cc/shakediy**

Top Tip

Lego case

The Raspberry Shake enclosure can be 3D-printed or laser-cut, but there is also a Lego case designed for the system. Take a look at the Lego build instructions to make your own case from bricks. **magpi.cc/shakelego**

used to level the device with the levelling feet. If you bought an official enclosure, it will come with a small spirit level on the base. Use a screwdriver with the levelling feet to ensure that the bubble in the spirit level is inside the black circle.

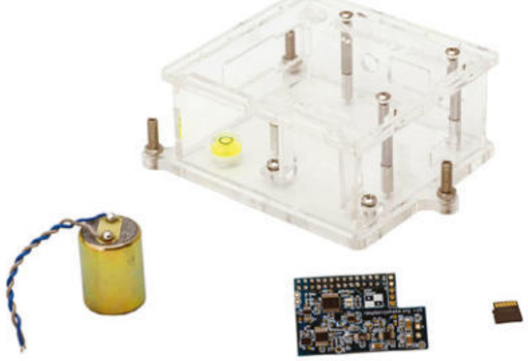
06 Position the Raspberry Shake

The device is designed to be left running 24 hours a day, monitoring for earth tremors. So you'll want to find somewhere out of the way. With the Raspberry Pi 3B included in the kit, you should use an Ethernet connection to the router, to avoid possible wireless LAN interference to the geophone (this is not an issue if using a Raspberry Pi Zero or 3B+ – see **magpi.cc/shakewlan**).

> The device is designed to be left running 24 hours a day, monitoring

You'll need to run an Ethernet wire directly from Raspberry Shake to your router. We used Devolo DLAN Powerline adapters (**devolo.co.uk**) to extend our Ethernet connection across the electrical wiring. We positioned our Raspberry Shake in the conservatory to the rear of our home.

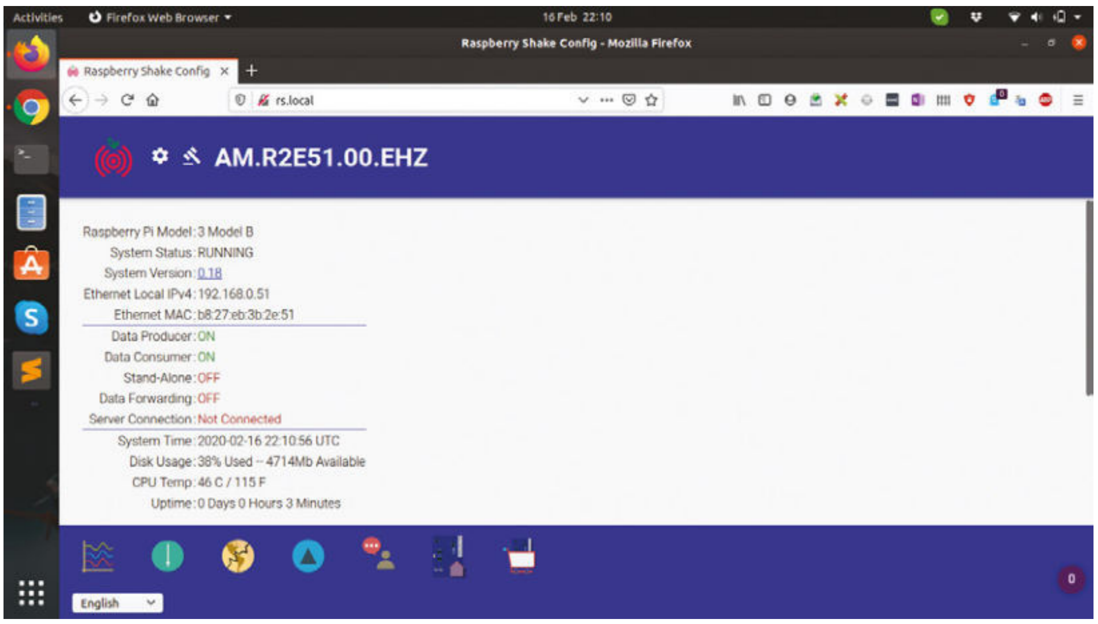
According to the makers of Shake: "For best results, install your Raspberry Shake on a bare floor (no carpet) and not on top of your desk. A good location for the Shake would likely be on the concrete slab of the lowest floor, near a foundation wall and away from furnaces, washing machines, air conditioners, and such."


The geophone, enclosure, Raspberry Shake board, and microSD card – you just need to add Raspberry Pi

07 Power up

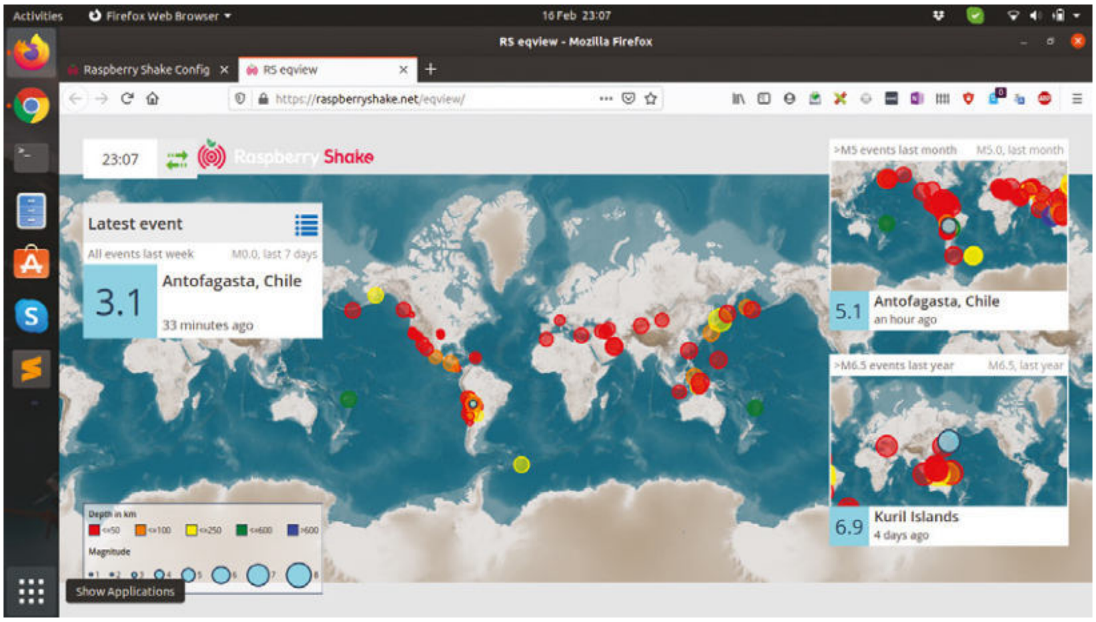
With Raspberry Shake in position and connected to your router, use the power adapter to turn on Raspberry Pi. A blue light will appear on top of the Raspberry Shake board.

You don't access Raspberry Pi directly with a keyboard and screen – instead, it is set up for remote connection over your network. Open a web



Raspberry Pi Model: 3 Model B
System Status: RUNNING
System Version: 0.18
Ethernet Local IPv4: 192.168.0.51
Ethernet MAC: b8:27:eb:3b:2e:51
Data Producer: ON
Data Consumer: ON
Stand-Alone: OFF
Data Forwarding: OFF
Server Connection: Not Connected
System Time: 2020-02-16 22:10:56 UTC
Disk Usage: 38% Used -- 4714Mb Available
CPU Temp: 46 C / 115 F
Uptime: 0 Days 0 Hours 3 Minutes

Once the Raspberry Shake device is set up, you can access its settings via a web interface on the local network



Raspberry Shake Earthquake View is used to track seismic activity detected by other users around the world

browser from another computer on the network, and go **http://rs.local** (don't forget the 'http://' part. (Note that 'rs.local' replaces the former 'raspberryshake.local'). You will see the Raspberry Shake web interface.

08 Change SSH and setup

The default SSH username and password are 'myshake' and 'shakeme'. Default SSH passwords are a security risk, so we're going to change it. Click the Actions icon near the top of the interface, then click on the Actions tab. Now click Change SSH Password. Enter the current password 'shakeme' and then your new password. Press **ENTER** to save the new password.

See 'Ready, Set, Get Hacked!' on the Raspberry Shake website for more information on security: **magpi.cc/shakehacked**.

09 Join the team

With your Raspberry Shake password changed, you can turn on data sharing and join the Raspberry Shake community. This enables you to share your data with the citizen science project.

Click Home and Settings. Fill out your details in the General section and click Set Location. The location data is randomised by a couple of hundred yards to preserve privacy (**magpi.cc/shakelocation**).

Finally, enter the floor that the device is on – this is zero-indexed, so 0 is the ground floor – and how many floors you have in the house.

Top Tip

Geophone datasheet

The RGI-4.5Hz geophone is the heart of the Raspberry Shake project. Other geophones are available, but you'll need this specific model if you want to join in the community project. To learn more about the geophone, take a look at this datasheet. **magpi.cc/geophone**

Now click the Data tab and tick the box marked 'Forward Data'. Read the licensing information and click Save and Restart. Raspberry Shake will restart (you may be prompted to enter your new SSH password from Step 08).

10 Earthquake and Station View

Now that you're part of the wider Raspberry Shake community, it's time to take a look at earthquake activity around the world. Click Raspberry Shake Earthquake View to see a global map. The circles indicate earthquake activity. The colour of the circle corresponds to its depth, with red circles showing it's closer to the Earth's surface. The size of the circle indicates its severity. Click on any circle to see more information.

If you want to see all the Raspberry Shake devices (including your own), make a note of your station number and click on the Station View icon. Here you will see all the devices running in the world. Click any device to view its data.

Top Tip

Video tutorial

Raspberry Shake has produced a series of video tutorials for the assembly and levelling of the device. **magpi.cc/shakestart**

Part 02

Configuring your magic mirror

Last month we built a mini magic mirror. Now it's time to upscale and take a tour of the powerful software that drives it

MAKER

PJ Evans

PJ is a writer, tinkerer, and enjoys installing Raspberry Pi Zero computers where no-one asked for them to be.

@mrpjevans

In *The MagPi* #90, we built a simple magic mirror using a semi-transparent piece of acrylic and a cheap picture frame. Mounting a Raspberry Pi screen behind it allows text to appear as if floating in air. We also introduced MagicMirror², an application solely designed for creating beautiful displays on your mirror. This month we're going to deep-dive into the software and see how we can customise our display. First of all, you may notice that we've given our mirror a bit of an upgrade, using a 23-inch monitor, semi-transparent film, and a larger Ikea Ribba frame. The software build is the same, just the hardware is bigger.

01 Orientate yourself

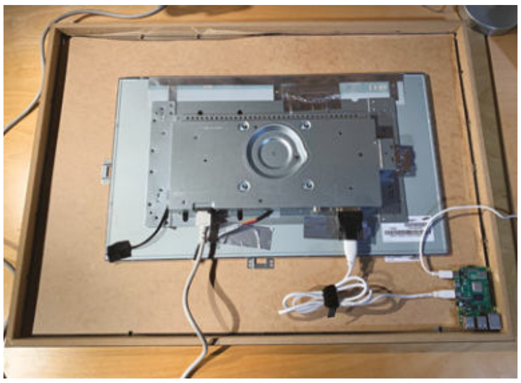
Can a mirror be upside-down? A magic mirror can! Most mirrors tend to be portrait, whereas screens are normally landscape. Normally we could make some changes to **/boot/config.txt** to easily rotate the screen, but with Raspberry Pi 4's fancy new graphics support, this is no longer possible. To rotate your display 90° so it fits your mirror, open a Terminal and enter the following:

```
sudo nano /etc/xdg/lxsession/LXDE-pi/
autostart
```

Now add this line to the end of the file:

```
@xrandr --output HDMI-1 --rotate right
```

Save the file (**CTRL+X**) and reboot. Your display should now be portrait.



The rear view shows the mounted monitor and Raspberry Pi computer. Removing the monitor's bezel gets the screen as close as possible to the surface

02 Safety first

Sadly, MagicMirror² doesn't (yet) come with an easy configuration utility. For now, you will need to do some text editing of **config.js**. Don't worry: as long as you make copies of your files, it's going to be hard to break anything. If MagicMirror² refuses to start, just copy the file back. Here's how it works from a Terminal:

```
cd ~/MagicMirror/config
cp config.js config.js.backup
```

If anything goes wrong, just copy the config backup back and try again:

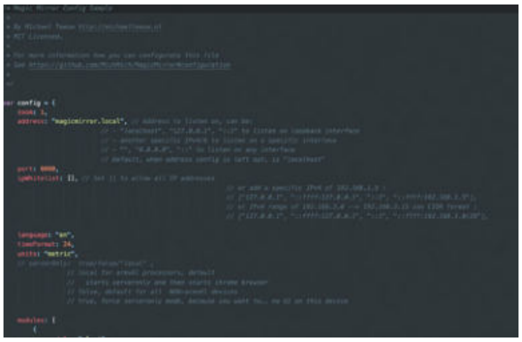
```
cp config.js.backup config.js
```

Each time you edit **config.js**, you'll need to restart MagicMirror² for changes to take effect. To do this at any time:

```
pm2 restart MagicMirror
```

You'll Need

- One-way mirror film **magpi.cc/onewayfilm**
- Monitor with bezel removed (We used a spare 23-inch screen)
- Ikea Ribba frames to fit monitor **magpi.cc/ribba2**
- Sample config.js and compliments.js files (optional) **magpi.cc/mm2git**



```
* Magic Mirror Config Sample
*
* By Michael Teeuw http://michaelteeuw.nl
* MIT Licensed.
*
* For more information how you can configurate this file
* See https://github.com/MichMich/MagicMirror#configuration
*
*/

var config = {
	zoom: 1,
	address: "magicmirror.local", // Address to listen on, can be:
	                      // - "localhost", "127.0.0.1", "::1" to listen on loopback interface
	                      // - another specific IPv4/6 to listen on a specific interface
	                      // - "", "0.0.0.0", "::" to listen on any interface
	                      // Default, when address config is left out, is "localhost"
	port: 8080,
	ipWhitelist: [], // Set [] to allow all IP addresses
	                                  // or add a specific IPv4 of 192.168.1.5 :
	                                  // ["127.0.0.1", "::ffff:127.0.0.1", "::1", "::ffff:192.168.1.5"],
	                                  // or IPv4 range of 192.168.3.0 --> 192.168.3.15 use CIDR format :
	                                  // ["127.0.0.1", "::ffff:127.0.0.1", "::1", "::ffff:192.168.3.0/28"],

	language: "en",
	timeFormat: 24,
	units: "metric",
	// serverOnly:  true/false/"local" ,
	           // local for armv6l processors, default
	           //   starts serveronly and then starts chrome browser
	           // false, default for all  NON-armv6l devices
	           // true, force serveronly mode, because you want to.. no UI on this device

	modules: [
		{
```

▲ Here's a section of the config file. It's JavaScript code so may look a little different if you're used to Python

03 Meet the config file

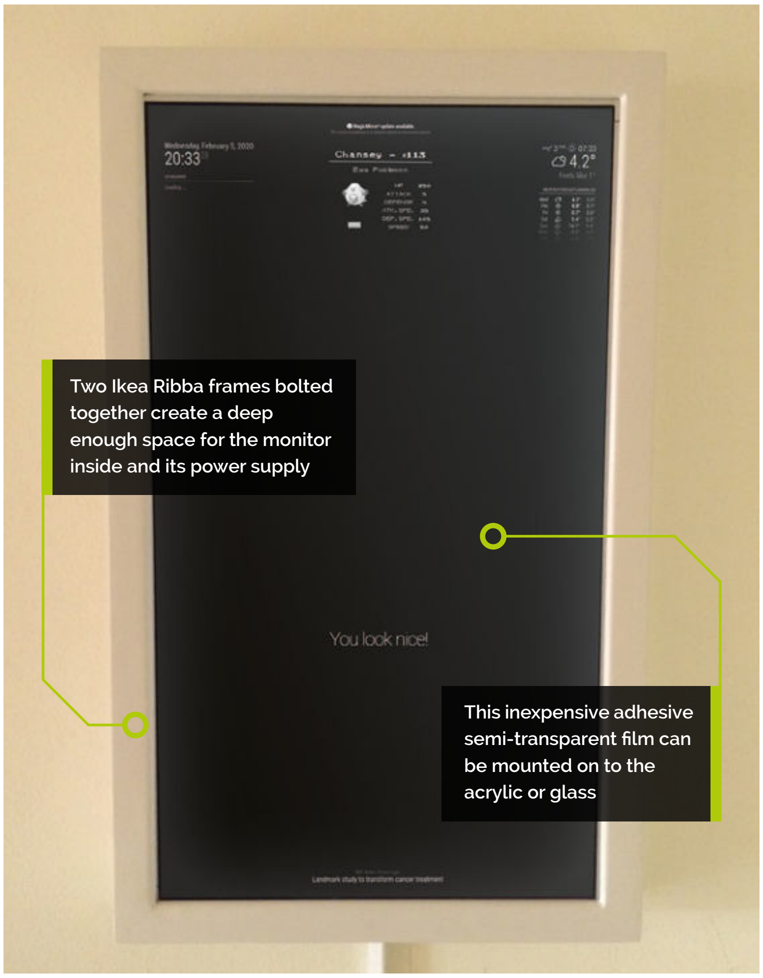
Your config file controls some of the fundamental features of your mirror, as well as the various modules. It is formatted in a JavaScript file. This is a very well-structured language but unforgiving. A misplaced '{' or '[' and nothing will work (hence the previous step). The **config.js.example** file is a great way to explore without changing anything. You'll see how to change the 'zoom' (text size), units (metric or imperial), and whether you want a 24-hour clock or not. Most importantly, the 'modules' section controls which modules (code that gives certain functionality) are loaded and where they're placed.

04 The default modules

MagicMirror² comes with a selection of modules pre-installed. Start by tailoring them to your specific needs. Find the 'modules' section in config.js. Within the two square brackets are sections contained within curly brackets: { }. Each one is a single module. Each module has different requirements but each one requires a 'module' line like this:

```
module: "name-of-module",
```

Most also require a 'position', which can be 'top_bar', 'top_left', 'lower_third', and many others. This controls where the module is displayed on the mirror. If a title is required, then 'title' allows you to change the text of the header. Finally 'config' will contain information that is specific to that module.



20:33
You look nice!
Two Ikea Ribba frames bolted together create a deep enough space for the monitor inside and its power supply
This inexpensive adhesive semi-transparent film can be mounted on to the acrylic or glass

05 Whatever the weather

Let's use the weather module as an example. Under 'modules', find 'weatherforecast'. You have the option to change the title (maybe 'Will it rain?') and the position of the module on the screen. In 'config' you have three pieces of information to complete. To get your own weather forecast, go to **openweathermap.org** and register for a free account. You can then create an API key (a secure way of your mirror communicating with the service), which you need to specify here in 'appid'. Change the name of your location as you wish and finally change the location ID to the correct one listed in this file: **magpi.cc/citylist**. Restart MagicMirror² and see your local weather!

06 Breaking (glass) news

The default news feed on the mirror comes from The New York Times, which may not be your cup of tea. The 'newsfeed' module works with any RSS feed, of which there are millions to choose

Top Tip

Learn JSON

If you're serious about configuring MagicMirror², knowing the JSON data format is essential. Have a look at **json.org**.

Top Tip

Back it all up

If you're editing a module or config file, it costs nothing to create a copy – then if it all goes wrong, you can try again.

from (and you can have multiple newsfeeds if you wish). Let's change the newsfeed to the BBC. Find the module 'newsfeed' and you'll see under 'config' the 'feeds'. This is surrounded with square brackets [ ], which means we can have multiple entries. Change 'New York Times' to 'BBC News' and the 'url' to 'http://feeds.bbci.co.uk/news/rss.xml'. Restart MagicMirror² and now you're getting the headlines from the UK.

07 Vampire mode (no reflection)

By now you may be finding it a little frustrating if you've already put up your mirror. With a bit of reconfiguration, you can access the mirror display using a web browser and work on it from your desktop. Edit **config.js** and have a look at the first few lines under 'config'. These control access to the display. It's locked down by default (which is good), but we can allow other computers access. Change these following lines as shown:

```
address: "",
ipWhitelist: []
```

This allows any IP address on your network to access the server. You'll need to restart MagicMirror² for changes to take effect. Now you should be able to see your display at http://<your mirror's IP address>:8080.

It may be easier to configure your screen before mounting it in the mirror

08 Client and server mode

The reason we can so easily see the display in a web browser is that MagicMirror² is split in to two parts: the client, the software that displays the screen; and the server, which generates the content. This clever split allows you to generate the content from a separate computer on the network, which is handy if you want to do something really intensive. It also allows you to have multiple mirrors that all show the same display, which makes rolling out changes really easy.

To start an installation of MagicMirror² without a display (server):

```
node serveronly
```

To create a client that gets its content from the server:

```
node clientonly --address <ip of server>
--port 8080
```

" In the centre of the screen, there are 'compliments' rotating every few seconds. You can change these "

09 Editing modules

Let's have a look at modifying existing modules. You'll have seen in the centre of the screen, there are 'compliments' rotating every few seconds. You can change these in **config.js**, but let's look in the actual code so we can get a feel of how things are arranged. In Terminal, navigate to where the default modules are installed:

```
cd ~/MagicMirror/modules/default
```

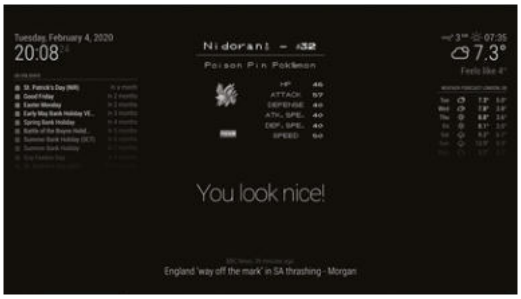
If you do an `ls` to get a directory listing, you'll be able to tell what's available. We're interested in the 'compliments' module:

```
cd compliments
```

Now edit the file:

```
nano compliments.js
```

Look at the various text strings and change them to whatever you like. Save the file and



Tuesday, February 4, 2020
20:08
Nidoran♀ – #32
Poison Pin Pokémon
HP 46
ATTACK 57
DEFENSE 40
ATK. SPE. 40
DEF. SPE. 40
SPEED 50
7.3°
Feels like 4°
St. Patrick's Day (NIR)
Good Friday
Easter Monday
Early May Bank Holiday VE...
Spring Bank Holiday
Battle of the Boyne Holid...
Summer Bank Holiday (SCT)
Summer Bank Holiday
You look nice!
England 'way off the mark' in SA thrashing - Morgan

There are many, many modules available for all kinds of uses. This one shows you a daily Pokémon!

restart MagicMirror² to see your new messages. A sample **compliments.js** file can be seen at **magpi.cc/mm2compliments**.

10 More modules

Great news: you are not restricted to the default modules. There are hundreds of community-built modules that are free to download and install. They cover all kinds of useful information, including stock prices, local transportation, prayer guides, and even how your local Minecraft server is holding up. Luckily, a directory of MagicMirror² modules is maintained on the main site's wiki: **magpi.cc/mirrormodules**.

Most modules will require some configuration, so make sure you look at the **README** file and follow the instructions carefully. There's no limit to how many modules you can have, bar the positions available on the screen.

11 Installing modules

Sadly, we don't have a nice package manager for MagicMirror², so installing modules tends to involve using Git to fetch the code. For an example, we're going to install 'Daily Pokemon'. From the Terminal, we'll go to the **modules** directory, then get the code from GitHub.

```
cd ~/MagicMirror/modules
git clone https://github.com/NolanKingdon/
MMM-DailyPokemon
cd MMM-DailyPokemon
npm install
```

The final command gets all the libraries that the module needs to run. Once completed, edit the config file:

Figure 1

DOWNLOAD THE FULL CODE: magpi.cc/mm2config

Language: **JSON**

```
modules: [
    {
            module: "MMM-DailyPokemon",
            position: "top_center",
            config: {
                updateInterval: 600000,
                minPoke: 4,
                maxPoke: 151,
                grayscale: true,
                showType: true,
                language: "en",
                genera: true,
                gbaMode: true,
                nameSize: 26
            }
    },
```

```
nano ~/MagicMirror/config.js
```

Create a new line after `modules: [` and add the code from the **Figure 1** listing (or use the download link for the full, edited **config.js** file). Make sure you end with a comma. Restart MagicMirror² and admire your daily Pokémon.

12 More modules

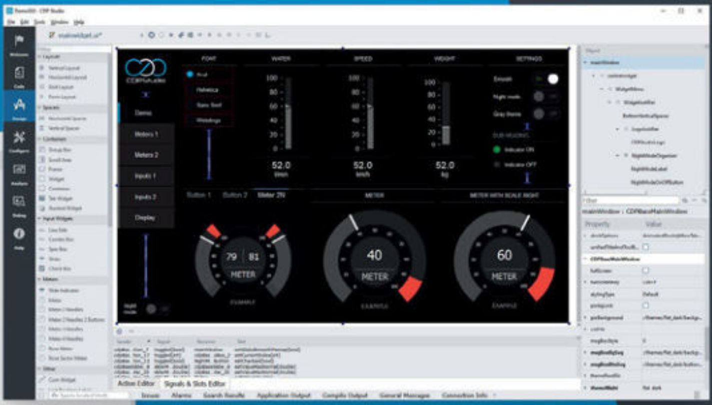
As we've already said, there's a dizzying array of modules to choose from, and you can even write your own. If you're in the mood for customising, here are a few of our favourites. You can find all of them, along with installation instructions at **magpi.cc/mirrormodules**.

magpi.cc/mmmstocks
Every information screen needs a stock ticker, right?

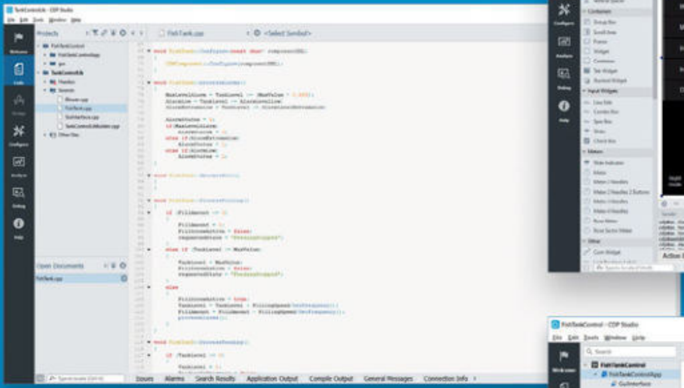
magpi.cc/mmmwiki
Up your knowledge as you get ready to leave the house with these random snippets from Wikipedia.

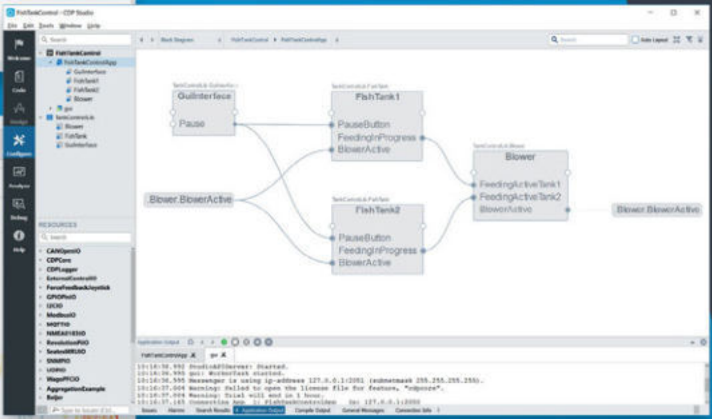
magpi.cc/mmmukrail
Is your train going to be on time? If you're in the UK, this module will let you know. Many other countries have equivalent modules too.

Design

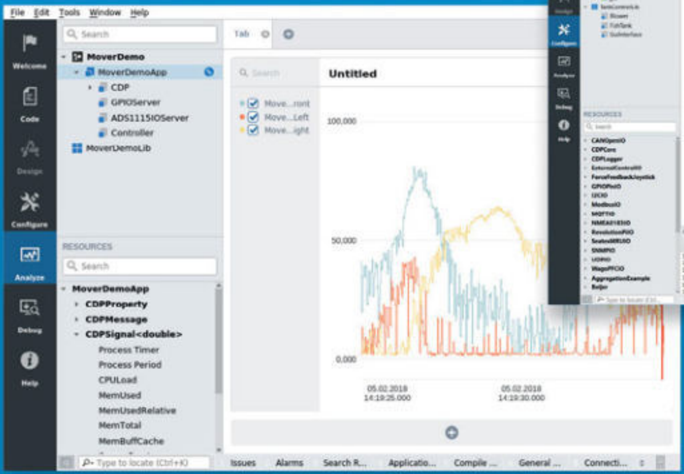


Code





Configure



Analyze

NEW GRAPHICAL EDITOR!

CDP STUDIO

Professional control system development tool

CDP Studio is a development platform for industrial control systems, now coming with a free version for non-commercial use. The system can run on a Raspberry Pi, supports C++, open source libraries and has a large feature toolbox including GPIO, I2C and MQTT. Its built in GUI design tool and features lets you code less and do more.

CDP Technologies AS
Nedre Strandgate 29, P.O. Box 144
NO-6001 Ålesund, Norway
Tel: +47 990 80 900 • info@cdptech.com
www.cdpstudio.com

CDPStudio

Part 02

Make beats in style with Hex-A-Pad

MAKER

Mike Cook

Veteran magazine author from the old days, writer of the Body Build series, plus co-author of *Raspberry Pi for Dummies*, *Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.

magpi.cc/mikecook

Hex-A-Pad is a stylish, capacitive touch sensor for triggering sounds. In this tutorial we add the ability to record a 16-step sequence, plus a MIDI option

Last month we looked at how to make the Hex-A-Pad, and we showed you simple software to trigger sound samples. This month we will look at software that makes better use of the LEDs, and allows the recording and playback of a 16-step sequence. Not only that, we will add a MIDI option and discuss the best way to arrange notes on this box.

01 Software concept

The idea of the software is that it acts as a control panel to select which bank of sounds the pad will produce, and which sequence recording mode to use, along with whether to use sound samples or MIDI messages. The idea being that the software will only be used to set up various modes of operation; after that, all playing is controlled by the Hex-A-Pad itself. On the Hex-A-Pad, there are eight touch-sensitive triggers: six on each the hexagon's sides and two in the middle. The sensors on the side always trigger notes, and the two in the middle can trigger notes as well, or record and play back a 16-step sequence. See **Figure 1**.

02 Trigger LEDs

Each of the eight touch sensors has an LED associated with it. In the code we presented last month, the LEDs turned on when a sensor was

You'll Need

- Hex-A-Pad capacitive touch sensor trigger **magpi.cc/90**
- MIDI sound generator (optional), as in **magpi.cc/63**
- USB to MIDI interface cable (optional) **magpi.cc/USB2MIDI**

Figure 1

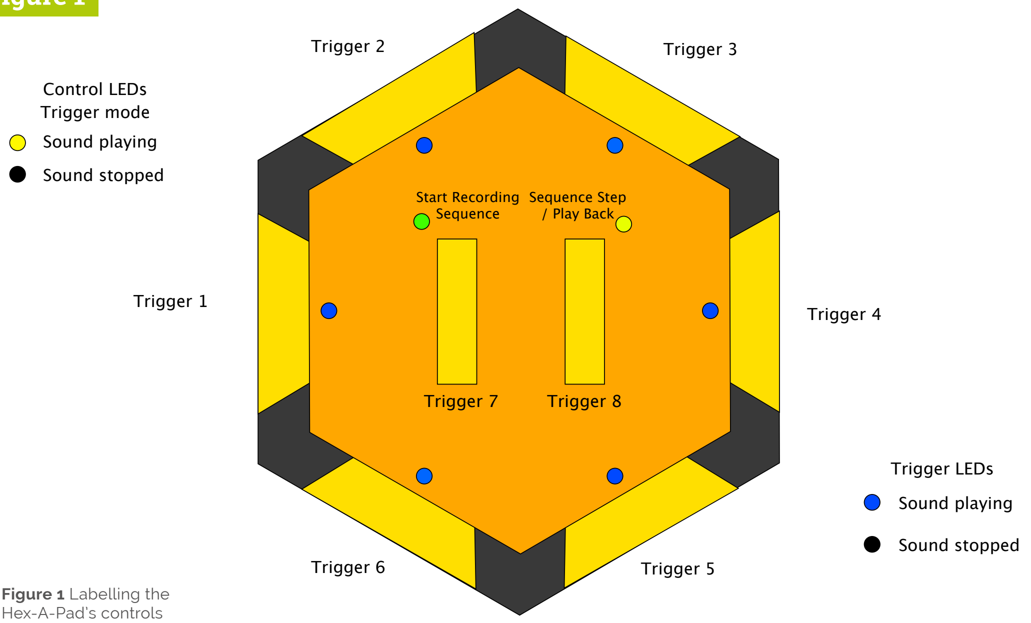


Figure 1 Labelling the Hex-A-Pad's controls

Top Tip

Why pentatonic?

A pentatonic scale is good for improvising, as simple runs of notes sound good. It has been independently developed in lots of cultures, and so lends itself to world (as well as Western) instruments. We use samples from a koto, a Japanese stringed instrument, for Bank 3 of the sound samples.

touched, triggering a sound, and turned off when the sensor became untouched. However, we felt a much more pleasing effect would be generated if the LED remained on for the duration of the sound. This could be done for samples because Pygame has a function to check if a sound channel is busy or not, providing you made a note of the channel being used when you set the sound playing. For MIDI notes, the LEDs are turned off when the MIDI note off message is sent.

03 Polyphony

Polyphony is the ability to play more than one note at a time, and the Pygame system (as well as MIDI) allows this. In the case of playing samples, the Pygame mixer can be assigned a channel for each sound you want to play at the same time. This can be the same sample triggered from the same sensor, or a different sample from the same bank. We set the number of channels to 20, but still check there is a free one before starting any sample – if not, a warning message is sent to the console.

04 Sequence recorder

Incorporated into the software is a 16-step sequence recorder. The current step in indicated by a virtual LED being lit on a ring. As we need to use the two central touch sensors for playback and recording, a sequence is only possible in the six-sample trigger mode. The sequence can be entered in two ways: live recording or single step mode. With live recording, once the recording control has been touched, and the red LED lights next to it, the recording only starts when the first note is triggered. After that, any notes played are entered into the current sequence step until it has gone through every step.

05 Quantised sequence

When recording a sequence live, the sequence is quantised. That means if you play three notes in rapid succession, then they will probably all end up in the same sequence position – and so, on playback, they will all be played at the same time. Each sequence step is capable of storing up to six notes, so you are not going to run out of depth

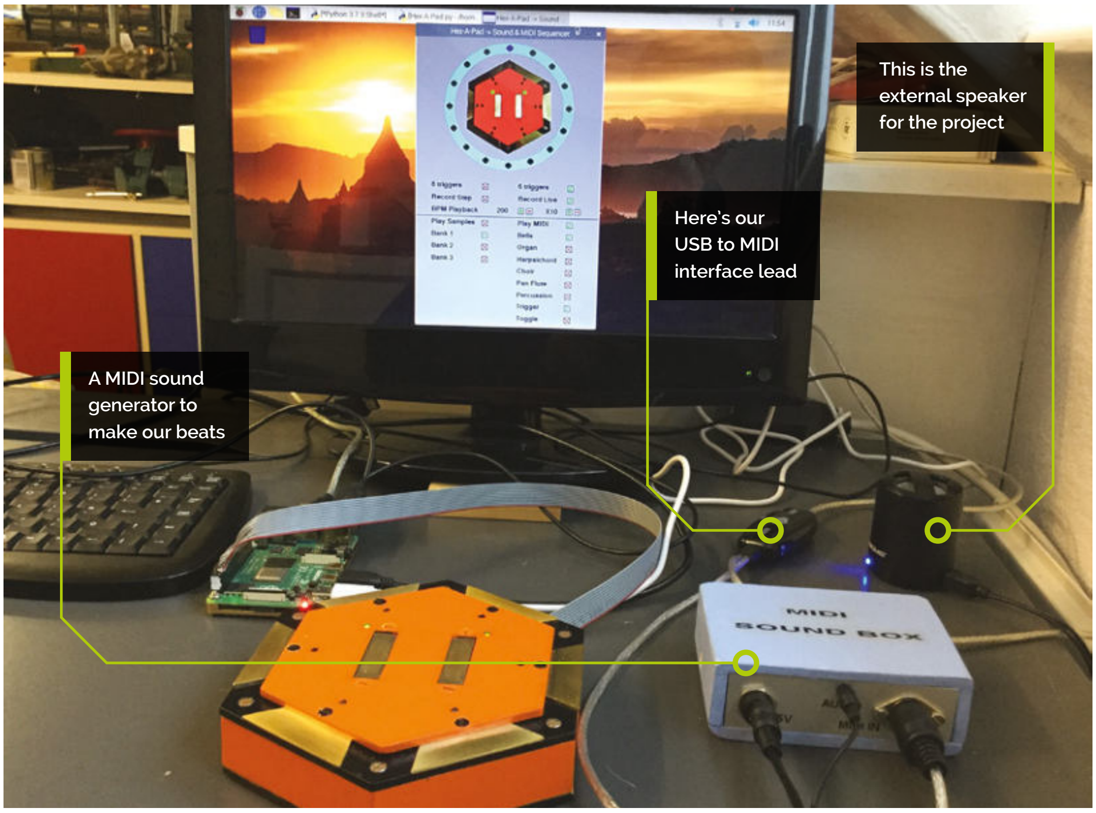

This is the external speaker for the project
Here's our USB to MIDI interface lead
A MIDI sound generator to make our beats
Hex-A-Pad – Sound & MIDI Sequencer
MIDI SOUND BOX

Figure 2

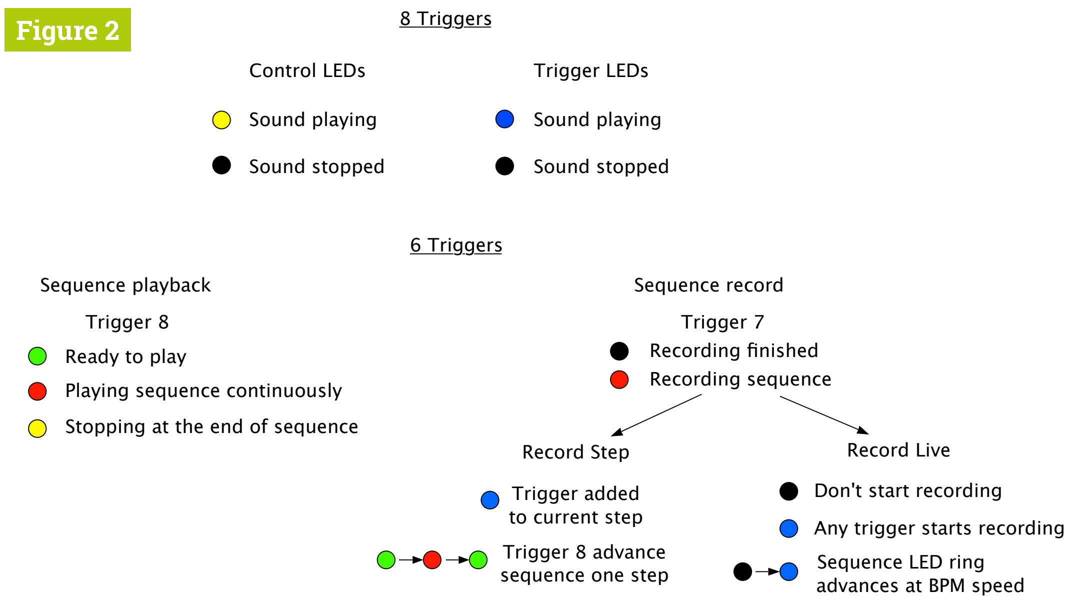


Figure 2 How the LEDs work in the different operating modes

Figure 3 Clock notation for generating scales and keys

in the sequence. Sometimes, it is preferable to be able to enter the sequence one note at a time to ensure the quantisation of the sequence is under your control.

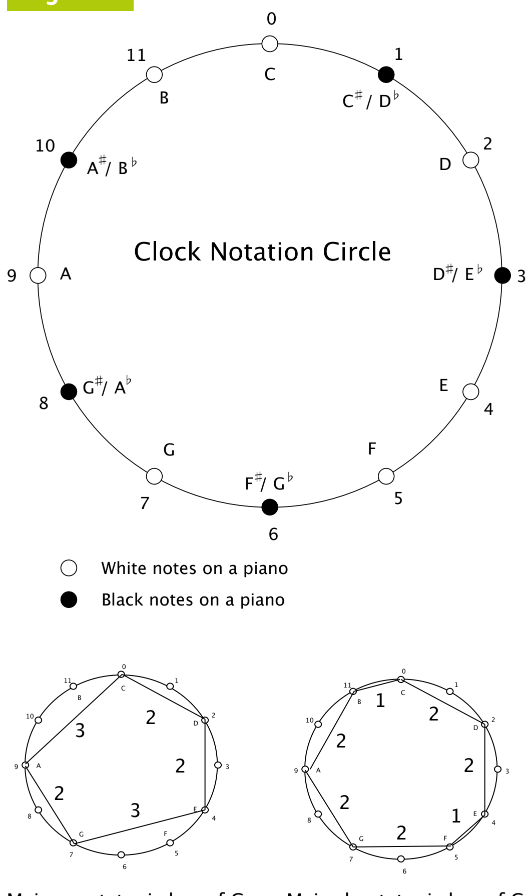
06 Step sequence

The step sequence mode of recording allows you to enter as many or few notes into each sequence step. In this mode, once the record sensor is touched, each note played is stored in the current sequence step, and the next step is selected by touching the right-hand control sensor. The progress of the lit LED around the sequence ring shows you the current position in the sequence. When the sequence is complete, the red recording LED goes out and you are ready to play the sequence using the right-hand control sensor. The sequence can be played back at any speed by altering the BPM (beats per minute) parameter.

07 Playing a sequence

The sequence playback can be in three states: not playing, playing, and stopping. The first two are self-explanatory and the control LED is either green or red. Once the sequence is playing, it will continue to loop round, playing it continuously. To stop a sequence playing, press the trigger control again and the LED will turn yellow/

Figure 3

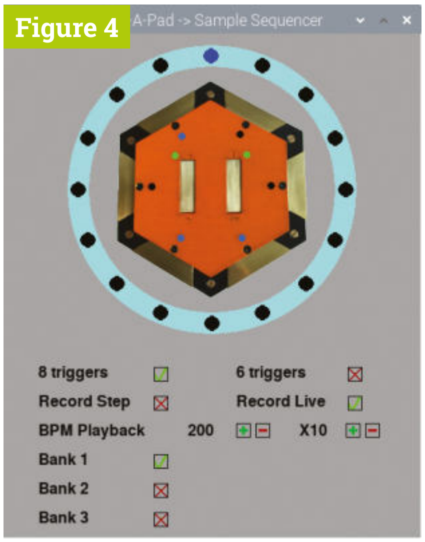


Top Tip

Toggle mode

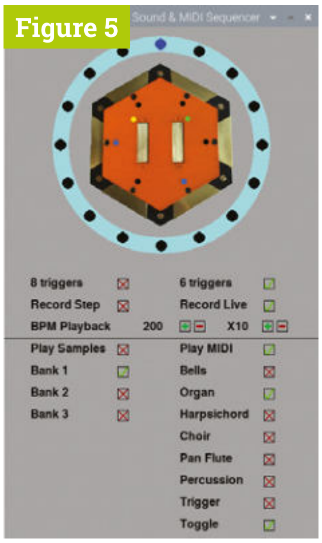
The MIDI toggle mode sends a note on message when touched, and a note off message when touched again. This is for triggering drone notes or ambient pads whose sound develops and changes over a long cycle.

Figure 4



8 triggers ✓ 6 triggers ☒
Record Step ☒ Record Live ✓
BPM Playback 200 + − X10 + −
Bank 1 ✓
Bank 2 ☒
Bank 3 ☒

Figure 5



8 triggers ☒ 6 triggers ✓
Record Step ☒ Record Live ✓
BPM Playback 200 + − X10 + −
Play Samples ☒ Play MIDI ✓
Bank 1 ✓ Bells ☒
Bank 2 ☒ Organ ✓
Bank 3 ☒ Harpsichord ☒
Choir ☒
Pan Flute ☒
Percussion ☒
Trigger ☒
Toggle ✓

Figure 4 Software screen for sample sounds

Figure 5 Software screen for sample sounds and MIDI output

orange. This indicates that the sequence will stop at the end of the current cycle. The use of the LEDs in recording a sequence is shown in **Figure 2** (previous page), but let us assure you that this looks far more complicated than it is: in real use, these controls quickly become natural.

08 Choosing notes

If you are using the Hex-A-Pad for just percussion, then you will not have the problem of choosing which six notes to trigger. However, if you do want to produce tunes, you have to pick the notes and a key. In Western music this is normally a heptatonic scale, i.e. an octave is split up into seven notes. We normally think of there being eight notes in an octave, but the first note and last note are the same note, just an octave apart. However, the six triggers on the Hex-A-Pad are ideal for another type of scale: a pentatonic scale.

“This looks far more complicated than it is: in real use, these controls quickly become natural”

09 Pentatonic scales

With a pentatonic scale, the octave is split into five notes; having the first and last notes the same but an octave apart gives us six notes in a scale. The first, or root note, of any scale gives it its key name. A simple example of a pentatonic scale is all the black notes on a piano; if we start with the F♯/G♭ and play just the black notes, we get the major pentatonic scale of F♯/G♭. One of the most popular major pentatonic scales is that of C, and we have used that to choose our notes.

10 Building a scale

The intervals, or distances between the notes, of a scale are expressed in semitones; there are twelve semitones in an octave. The notes in the major pentatonic key of C are C, D, E, G, A, and back to C, so these can be represented as intervals or number of semitones between notes as 2 | 2 | 3 | 2 | 3. **Figure 3** shows the circular or clock method of representing scales: each hour position is one semitone away from the next, so it's easy to generate a key by starting at a point and using the sequence of intervals for the scale you want. Note that the shape made by joining the notes of a pentatonic scale is a pentagon. Starting at any other note and keeping the same sequence of intervals just rotates this figure and changes the key.

11 The software

We have written two pieces of software for driving the Hex-A-Pad: one that handles only sound samples and the other that includes MIDI note generation as well. The screen displays of each are shown in **Figures 4** and **5**. The sound-sample-only version is shown in the **hexapadsamples.py** listing, and both are available on our GitHub pages. They run under the Pygame framework, and require the MPR121 touch sensor library we used in last month's code.

12 Customisation

Changing the sound samples and MIDI mapping is the most useful way in which you can customise the instrument. Samples should be added to the **sounds** directory, and their new names included in the declaration of the sound banks. Similarly for MIDI, the note and velocity values are in the declaration of the five `mMap` lists, with the different instrument names and program change numbers in the `mInstName` and `mInst` lists respectively. You can explore different sounds and scales. The use of MIDI requires you have a USB-to-MIDI interface cable and a MIDI sound generator.

hexapadsamples.py

▸ Language: **Python**

DOWNLOAD THE FULL CODE:

magpi.cc/pibakery

```python
#!/usr/bin/env python3
# Hex-A-Pad Sample player & sequence
# By Mike Cook January 2020

import sys
import time
import os
import board
import busio
import adafruit_mpr121
import digitalio as io
import pygame
import math

def main():
    global last_touch
    init()
    setMPR121()
    last_touch = cap.touched()
    print ('Hex-A-Pad sample player')
    drawFixed()
    while True :
        checkForEvent()
        checkPlaying()
        if playback : checkPlayback(False)
        if not capSenseNew.value:
            cur_touch = cap.touched()
            for i in range(0, 8):
                readPins(i, cur_touch)
            last_touch = cur_touch

def readPins(i, c_touch) :
    global last_touch, playing, contPlay, recording
    pin_bit = 1 << i
    if c_touch & pin_bit and not last_touch & pin_bit:
      if i < 6 : LEDs[i].value = True
      if choices[0] : # 8 triggers
          if i == 6 : controlLEDs(0, 3)
          if i == 7 : controlLEDs(1, 3)
      if pygame.mixer.find_channel() :
          if choices[0] or i <6 :
              playing[i] = sounds[bank][i].play()
              if recording and i < 6 : saveNote(i)
              if i == 6 : contPlay[0] = True
              if i == 7 : contPlay[1] = True
              updateTrigger(i, True)
          elif choices[5] :
              if i == 6 : startRecording()
              if i == 7 : controlPlayback()
      else :
          print("channel overflow")

def saveNote(note) :
    global savePos, stepCount, playback
    if not playback : playback = True
    savePos += 1
    savePos = constrain(savePos, 0, 5)
    seqNotes[(stepCount + 1) & 0x0F][savePos] = note

def startRecording() :
    global stopping, playback, recording, savePos
    if not recording: # start recording
        wipeSequence()
        savePos = -1
        recording = True
        stopping = True # once round
        controlLEDs(0, 1)

def controlPlayback() :
    global stopping, playback, recording, nextStep, stepCount
    if recording and choices[1]: # step recording
        controlLEDs(1, 1)
        checkPlayback(True) # next step irrespective of time
        time.sleep(0.2) ; controlLEDs(1, 2)
    else :
        if playback :
            stopping = True
            controlLEDs(1, 3)
        else :
            nextStep = time.time()
            stepCount = -1
            controlLEDs(1, 1)
            playback = True

def checkPlayback(go) :
    global nextStep, stepCount, playback, stopping
    global recording, savePos
    if recording and choices[1] :
        nextStep = time.time() + 2.0
    if go : nextStep = 0.0
    if time.time() >= nextStep: # time for a new step
        nextStep = time.time() + stepTime
        stepCount = (stepCount + 1) & 0xF # 0 to 15
        updateStep(stepCount)
        if recording :
            savePos = -1
        else :
            if pygame.mixer.find_channel() :
                for i in range(0, 6) :
                    if seqNotes[stepCount][i] != -1 :
                        sounds[recordedBank][seqNotes[stepCount][i]].play()
        if stepCount == 15 and stopping :
            playback = False
            stopping = False
            recording = False
            controlLEDs(0, 2)
            controlLEDs(1, 2)
```

hexapadsamples.py

▶ Language: **Python**

```python
def updateTrigger(i,on):
    if i < 6 : col = (0,97,255)
    else : col = (255, 222, 0)
    if on :
        pygame.draw.circle(screen, col, dLED[i], 4, 0)
    else :
        pygame.draw.circle(screen, black,
                           dLED[i], 4, 0)
    pygame.display.update()

def updateChoices():
    for i in  range(0, len(choices)) :
        pygame.draw.rect(screen, backCol,
                         choiceRect[i], 0)
        pygame.draw.rect(screen, black,
                         choiceRect[i], 1)
        if choices[i] :
            screen.blit(yes, (choiceRect[i].left,
                              choiceRect[i].top))
        else :
            screen.blit(no, (choiceRect[i].left,
                             choiceRect[i].top))
    for i in range(0,len(incRect)) :
        pygame.draw.rect(screen, backCol,
                         incRect[i], 0)
        pygame.draw.rect(screen, black,
                         incRect[i], 1)
        if i & 1 :
            screen.blit(minus, (incRect[i].left,
                                incRect[i].top))
        else :
            screen.blit(plus, (incRect[i].left,
                               incRect[i].top))

    pygame.draw.rect(screen, backCol, ((188, 400),
                                      (32, 16)), 0)
    drawWords(str(bpm), 190, 400, black, backCol)
    pygame.display.update()

def drawFixed() :
    screen.fill(backCol)
    pygame.draw.circle(screen, (138, 229, 229),
                (sWide // 2, padCo[1] + 113), 149, 0)
    pygame.draw.circle(screen, backCol, (sWide // 2,
                            padCo[1] + 113), 127, 0)
    screen.blit(pad,padCo)
    for i in range(0,8) :
        pygame.draw.circle(screen, black,
                           dLED[i], 4, 0)
    for i in range(0,16) :
        pygame.draw.circle(screen, black,
                           sLED[i], 8, 0)
    pygame.draw.circle(screen, (0, 0, 255),
                       sLED[12], 8, 0)
    drawWords("8 triggers", 36, 340, black, backCol)
    drawWords("6 triggers", 240, 340, black, backCol)
    drawWords("Record Step", 36, 370, black, backCol)
    drawWords("Record Live", 240, 370, black, backCol)
    drawWords("BPM Playback", 36, 400, black, backCol)
    drawWords("X10", 305, 400, black, backCol)
    drawWords("Bank 1", 36, 430, black, backCol)
    drawWords("Bank 2", 36, 460, black, backCol)
    drawWords("Bank 3", 36, 490, black, backCol)
    controlLEDs(0, 2) ; controlLEDs(1, 2)
    updateChoices()

def updateStep(s) :
    s = (s - 3) & 15
    scol = (0, 0, 255)
    pygame.draw.circle(screen, scol, sLED[s], 8, 0)
    s -= 1
    if s < 0 : s = 15
    pygame.draw.circle(screen, black, sLED[s], 8, 0)
    pygame.display.update()

def drawWords(words,x,y,col,backCol) :
    textSurface = font.render(words, True,
                              col, backCol)
    textRect = textSurface.get_rect()
    textRect.left = x # right for align right
    textRect.top = y
    screen.blit(textSurface, textRect)
    return textRect

def init():
    global i2c, cap, capSenseNew, LEDs, sounds
    global pygame, soundNames, playing, contPlay
    global pad, sWide, sHigh, screen, backCol, seqNotes
    global padCo, dLED, sLED, black, font, ccol
    global choiceRect, yes, no, choices, incRect
    global plus, minus, bpm, stepTime, playback
    global nextStep, recording, stepCount, savePos
    global bank, recordedBank, stopping
    i2c = busio.I2C(board.SCL, board.SDA)
    cap = adafruit_mpr121.MPR121(i2c)
    capSenseNew = io.DigitalInOut(board.D4)
    capSenseNew.direction = io.Direction.INPUT
    capSenseNew.pull = io.Pull.UP
    LEDs = []
    LEDpin = [board.D17, board.D18, board.D27,
              board.D22, board.D23, board.D24]
    for i in range(0, len(LEDpin)):
        led = io.DigitalInOut(LEDpin[i])
        led.direction = io.Direction.OUTPUT
        LEDs.append(led)
    cap.reset()
    i2c.scan() # this is needed
    # Initialise Pygame system
    pygame.mixer.quit()
    pygame.mixer.pre_init(44100, -16, 2, 512)
    pygame.init()
    pygame.mixer.music.set_volume(1.0)
    pygame.mixer.set_num_channels(20)
    pygame.init()
    pygame.display.set_caption(
        "Hex-A-Pad -> Sample Sequencer")
    os.environ['SDL_VIDEO_WINDOW_POS'] = 'center'
    pygame.event.set_allowed(None)
    pygame.event.set_allowed([pygame.KEYDOWN,
                pygame.QUIT, pygame.MOUSEBUTTONDOWN,
                              pygame.MOUSEBUTTONUP])
    sWide = 430 ; sHigh = 520
    padCo = (sWide//2 - 100, 48) # top of screen
    screen = pygame.display.set_mode([sWide,
                                      sHigh], 0, 32)
    # Load graphics
    pad = pygame.image.load(
        "images/pad.png").convert_alpha()
    backCol = (160, 160, 160) ; black = (0, 0, 0)
    ccol = [ black, (255, 0, 70), (0, 255, 0),
             (255, 222, 0) ]
    x = [38, 69, 129, 161, 131, 68, 62, 136]
    y = [110, 58, 56, 109, 163, 163, 78, 77]
    dLED = []
    for i in range(0, 8):
        dLED.append( (x[i] + padCo[0],
                      y[i] + padCo[1]))
    sLED = [] ; s = math.radians(22.5)
    for i in range(0, 16):
        sLED.append((int(138 * math.cos(i * s) +
                         (sWide // 2)),
                     int(138 * math.sin(i * s) +
                         padCo[1] + 113)))
    choiceRect = [pygame.Rect((0, 0), (15, 15))] * 7
    for i in range(0,5) :
        ost = 0
        if i > 1 : ost = 30
        choiceRect[i] = pygame.Rect((155,
                        ost + 344 + i * 30), (15, 15))
    choiceRect[5] = pygame.Rect((355, 344), (15, 15))
    choiceRect[6] = pygame.Rect((355, 374), (15, 15))
    incRect = [pygame.Rect((240, 402), (15, 15))] * 4
    incRect[1] = pygame.Rect((260, 402), (15, 15))
    incRect[2] = pygame.Rect((353, 402), (15, 15))
    incRect[3] = pygame.Rect((373, 402), (15, 15))
    no = pygame.image.load(
        "images/0.png").convert_alpha()
    yes = pygame.image.load(
        "images/1.png").convert_alpha()
    plus = pygame.image.load(
        "images/plus.png").convert_alpha()
    minus = pygame.image.load(
        "images/minus.png").convert_alpha()
    choices = [False] * 7 ;
    presetC = [5, 2, 6] # power up choices
    for i in range(0,len(presetC)) :
        choices[presetC[i]] = True
    # Load Sounds three banks
    soundNames1 = ["tabla_ghe4", "bass_voxy_hit_c",
            "drum_splash_hard", "drum_tom_hi_hard",
            "drum_tom_lo_hard", "drum_snare_hard",
            "bass_voxy_c", "loop_amen_full"]
    sounds1 = [ pygame.mixer.Sound("sounds/"+
               soundNames1[i]+".wav")
               for i in range(0, len(soundNames1))]
    soundNames2 = ["Sax_a", "Sax_c", "Sax_d", "Sax_e",
                "Sax_g", "Sax_a2", "Sax_c2", "Sax_d2" ]
    sounds2 = [ pygame.mixer.Sound("sounds/"+
               soundNames2[i]+".wav")
               for i in range(0, len(soundNames2)) ]
    soundNames3 = ["K_a", "K_c", "K_d", "K_e",
                   "K_g", "K_a2", "K_c2", "K_d2" ]
    sounds3 = [ pygame.mixer.Sound("sounds/"+
               soundNames3[i]+".wav")
               for i in range(0, len(soundNames3))]
    sounds = [ sounds1, sounds2, sounds3]
    font = pygame.font.Font(None, 24)
    playing = [pygame.mixer.Channel(0)]*8
    contPlay = [False, False] ; playback = False
    recording = False ; nextStep = time.time()
    stepCount = -1 ; savePos = 0 ; bank = 0
    seqNotes = [] ; recordedBank = 0 ; stopping = False
    for i in range(0, 16) :
        seqNotes.append([-1, -1, -1, -1, -1, -1])
    bpm = 200 ; stepTime = 1 / (bpm / 60)

def wipeSequence() :
    global seqNotes, recordedBank
    for i in range(0, 16) :
        for j in range(0, 6) :
            seqNotes[i][j] = -1
    recordedBank = bank

def controlLEDs(led, col) :
    pygame.draw.circle(screen, ccol[col],
                       dLED[led + 6], 4, 0)
    pygame.display.update()
    mask = 0x30
    if led == 1 : mask = mask <<  2
    i2c.writeto(0x5A, bytes([0x79, mask])) # LEDs off
    if col == 1 : mask = 0x20
    if col == 2 : mask = 0x10
    if col == 3 : mask = 0x30
    if led == 1 : mask = mask <<  2
    if col != 0 :
        i2c.writeto(0x5A, bytes([0x78, mask]))
```

hexapadsamples.py

▶ Language: **Python**

```python

def checkPlaying(): # so we can turn off the lEDs
    global contPlay
    for i in range(0,6) :
        if (not playing[i].get_busy()) and LEDs[i].value :
            LEDs[i].value = False
            updateTrigger(i, False)
    if (not playing[6].get_busy()) and contPlay[0] :
        contPlay[0] = False
        controlLEDs(0, 0) # LED off
        updateTrigger(6, False)
    if (not playing[7].get_busy()) and contPlay[1] :
        contPlay[1] = False
        controlLEDs(1, 0) # LED off
        updateTrigger(7, False)

def setMPR121(): # top 4 sensor inputs to GPIOs
    # turn off cap sense
    i2c.writeto(0x5A, bytes([0x5e, 0]))
    #gpio enable top 4 bits
    i2c.writeto(0x5A, bytes([0x77, 0xf0]))
    # control 0 control 1 direction
    i2c.writeto(0x5A, bytes([0x73, 0xf0]))
    i2c.writeto(0x5A, bytes([0x74, 0xf0]))
    i2c.writeto(0x5A, bytes([0x76, 0xf0]))
    # limit sensor to first 8
    i2c.writeto(0x5A, bytes([0x5e, 8]))

def handleMouse(pos): # look at mouse down
    global choices
    #print(pos)
    for i in range(0,len(choices)) :
        if choiceRect[i].collidepoint(pos) and not choices[i]:
            pygame.draw.rect(screen, (192, 192, 0),
                             choiceRect[i], 0)
            pygame.display.update()
            if i == 6 or i == 7 : wipeSequence()
    for i in range(0,len(incRect)) :
        if incRect[i].collidepoint(pos) :
            pygame.draw.rect(screen, (192, 0, 192),
                             incRect[i], 0)
            pygame.display.update()

def handleMouseUp(pos): # look at mouse up
    global choices, bank
    for i in range(0,len(choices)) :
        if choiceRect[i].collidepoint(pos) :
            if not choices[i]:
                doRadioButton(i)
                choices[i] = True
    for i in range(0,len(incRect)) :
        if incRect[i].collidepoint(pos) :
            updateBPM(i)
    j = 0
    for i in range(2, 5) :
        if choices[i] : bank = j
        j += 1
    updateChoices()

def updateBPM(i) :
    global bpm, stepTime
    if i == 0 : bpm += 1
    if i == 1 : bpm -= 1
    if i == 2 : bpm += 10
    if i == 3 : bpm -= 10
    bpm = constrain(bpm, 30, 500)
    stepTime = 1 / (bpm / 60)

def constrain(val, min_val, max_val) :
    return min(max_val, max(min_val, val))

def doRadioButton(i) :
    group = [ [0, 5], [1, 6], [2, 3, 4] ]
    for k in range(0,len(group) ) :
        if i in group[k] :
            for j in range(0,len(group[k])) :
                choices[group[k][j]] = False
    if i == 0:
        controlLEDs(0, 0) # sequence LEDs off
        controlLEDs(1, 0)
    if i == 5 :
        controlLEDs(0, 2) # sequence LEDs green
        controlLEDs(1, 2)

def terminate(): # close down the program
    pygame.quit() # close pygame
    os._exit(1)

def checkForEvent(): # see if we need to quit
    event = pygame.event.poll()
    if event.type == pygame.QUIT :
         terminate()
    if event.type == pygame.KEYDOWN :
       if event.key == pygame.K_ESCAPE :
          terminate()
    if event.type == pygame.MOUSEBUTTONDOWN :
        handleMouse(pygame.mouse.get_pos())
    if event.type == pygame.MOUSEBUTTONUP :
        handleMouseUp(pygame.mouse.get_pos())

if __name__ == '__main__':
    main()
```

Part 11

C programming quick reference

Make use of these handy cheat sheets

MAKER

Simon Long

Simon Long is a software engineer working for Raspberry Pi, responsible for the Raspberry Pi Desktop on both Raspbian and Debian.

rpf.io

Control Structures

If

```
if (<test>)
  <code executed if test is true>
```

If-else

```
if (<test>)
  <code executed if test is true>
else
  <code executed if test is false>
```

Multiple if-else

```
if (<test1>)
  <code executed if test1 is true>
else if (<test2>)
  <code executed if test1 is false and test2
      is true>
else
  <code executed if test1 is false and test2
      is false>
```

Switch

```
switch (<variable>)
{
  case <testval1> : <code executed if
                      variable is testval1>
                    break;
  case <testval2> : <code executed if
                      variable is testval2>
                    break;

  default :         <code executed if
                    variable is neither
                      testval1 nor testval2>
                    break;
}
```

Switch with fall-through

```
switch (<variable>)
{
  case <testval1> : <code executed if
                      variable is testval1>

  case <testval2> : <code executed if
                      variable is either
                      testval1 or testval2>
                    break;

  default :         <code executed if
                      variable is neither
                      testval1 nor testval2>
                    break;
}
```

While

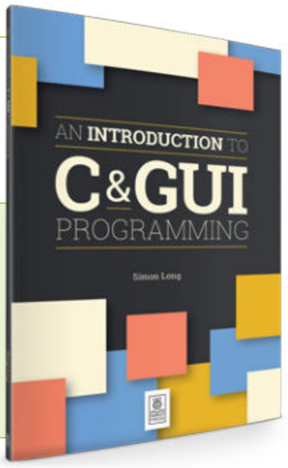
```
while (<test>)
  <code executed repeatedly while test is true>
```

Do-while

```
do
  <code executed once and then repeatedly
    while test is true>
while (<test>);
```

For

```
for (<initial condition>; <increment>;
        <termination condition>)
      <code executed repeatedly until
        termination condition is true>
```



In all loops, the keyword `break` can be used to exit the loop and resume execution immediately after the loop. In all loops, the keyword `continue` can be used to skip code remaining in the body of the loop and resume execution at the next iteration of the loop test.

Variable Types

Name	Description	Size (bytes)
`char`	Single alphanumeric character	1
`signed char`	Signed 8-bit integer (-128 – 127)	1
`unsigned char`	Unsigned 8-bit integer (0 – 255)	1
`short, signed short`	Signed 16-bit integer (-32768 – 32767)	2
`unsigned short`	Unsigned 16-bit integer (0 – 65535)	2
`int, signed int`	Signed 32-bit integer (-2147483648 – 2147483647)	4
`unsigned int`	Unsigned 32-bit integer (0 – 4294967295)	4
`long, signed long`	Signed 32-bit integer (-2147483648 – 2147483647)	4
`unsigned long`	Unsigned 32-bit integer (0 – 4294967295)	4
`float`	Floating-point value (+/- 3.402823 × 10^{38})	4
`double`	Double-precision floating-point value (+/- 10^{308})	8

Depending on platform, `int` can be either a `short int` (16 bits) or a `long int` (32 bits); on Raspbian, as per the table above, `int` is a long (32-bit) integer value.

Format Specifiers

Specifier	Format / type
`%c`	Alphanumeric character / char
`%d`	Signed decimal value / int
`%ld`	Signed decimal value / long int
`%u`	Unsigned decimal value / int
`%lu`	Unsigned decimal value / long int
`%o`	Octal value / int
`%lo`	Octal value / long int
`%x, %X`	Hexadecimal value / int [1]
`%lx, %lX`	Hexadecimal value / long int [1]
`%f`	Floating-point value / float
`%e`	Exponential value / float
`%s`	Text string / char pointer

1. `%x` displays a value as hexadecimal with lower-case letters a through f; `%X` displays it with upper-case letters A through F.

The width (or minimum number of characters printed) can be set by inserting a number between the `%` and the letter; this will pad a value shorter than this with spaces at the start. To pad with spaces at the end, insert a – between the `%` and the number. To pad with leading zeroes, insert a `0` between the `%` and the number.

For example, to print an integer variable with the value 42, using the format specifier `"%5d"` will print 42 with three spaces before it. The format specifier `"%-5d"` will print 42 with three spaces after it. The format specifier `"%05d"` will print it was 00042.

The number of decimal places shown for a floating-point or exponential value can be set by inserting a decimal point followed by a number between the `%` and the letter; this can be combined with a width by putting the width before the decimal point.

For example, to print a floating-point variable with the value 76.54321, using the format specifier `"%.2f"` will print it as 76.54. The format specifier

`"%08.2f"` will print it as 00076.54. (Note that the decimal point takes up one character of the specified width.)

Operators

The operators in the table below produce a result which can be assigned to another variable, e.g. `c = a + b`, but do not affect the values of `a` or `b`.

Symbol	Function
`a + b`	Addition
`a - b`	Subtraction
`a * b`	Multiplication
`a / b`	Division
`a % b`	Modulo (remainder of **a** / **b**)
`a & b`	Bitwise AND
`a | b`	Bitwise OR
`a ^ b`	Bitwise XOR
`a << b`	Bit shift left
`a >> b`	Bit shift right
`~a`	Bitwise 1's complement
`!a`	Logical NOT

The operators in the table below modify the value of `a` directly.

Symbol	Function
`a++`	Increment **a** by one [2]
`a--`	Decrement a by one [2]
`++a`	Increment **a** by one [2]
`--a`	Decrement **a** by one [2]
`a += b`	Increment **a** by **b**
`a -= b`	Decrement **a** by **b**
`a *= b`	Multiply **a** by **b**
`a /= b`	Divide **a** by **b**
`a %= b`	**a** = remainder of **a** / **b**
`a &= b`	Bitwise AND **a** with **b**
`a |= b`	Bitwise OR **a** with **b**
`a ^= b`	Bitwise XOR **a** with **b**
`a <<= b`	Bit shift **a** left by **b**
`a >>= b`	Bit shift **a** right by **b**

"Operators in the first table produce a result which can be assigned to another variable, but do not affect the values of a or b"

2. The difference between a++ and ++a is that if they are used in a test, such as `if (a++)`, a++ tests the value and then increments it, while ++a increments the value first and then tests the incremented value.

The operators in the table below are used for comparisons in tests.

Symbol	Function
`==`	Is equal to
`!=`	Is not equal to
`>`	Is greater than
`<`	Is less than
`>=`	Is greater than or equal to
`<=`	Is less than or equal to

Starter Electronics

WITH RASPBERRY PI

If you bought a Raspberry Pi to learn how to make electronic gizmos, you're in for a treat. Here are some tips to get started. By **Mark Vanstone**

Over the past few years, Raspberry Pi has gone from a small experimental computer to a very capable all-round system, while still retaining its small footprint. It has embedded itself in the hearts and minds of makers, coders, educators, inventors, and electro-artists around the world. Its flexibility and computing power now makes it the go-to platform for embedded systems (like smart devices) and workhorse tasks such as controlling other network systems or even robotics. The range of tasks Raspberry Pi can achieve becomes wider all the time as the community creates new ways to use it.

Gateway to the new frontier

Raspberry Pi has made a huge impact, not only on computer science education but also in the realm of makers and inventors. As well as being a really good-value computer, it gives the owner endless possibilities of connecting other electronics to it to produce useful gadgets and impressive demonstrations.

Raspberry Pi is your gateway to the new frontier of creative technology. If you can master the basics of how to connect electronics to a Raspberry Pi, you can use the same techniques to start inventing your own gadgets and electronic tools. If these are your first steps into electronics, you have a very exciting journey ahead.

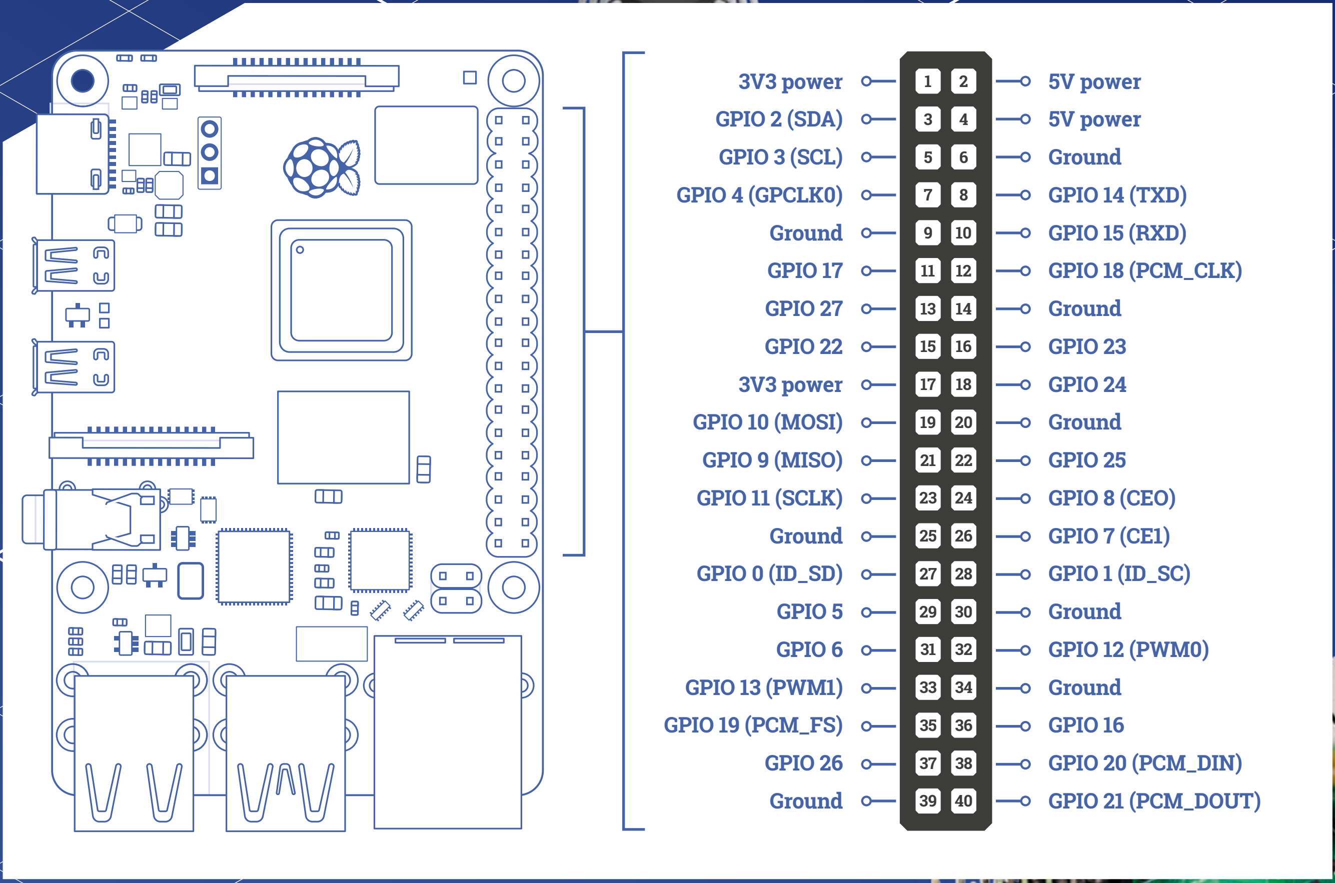
Expand and experiment

In the early days of home computers, most models had expansion ports of one sort or another and, very often, upgrading them involved opening up the case and soldering new circuits, or at least plugging in chips into empty slots. These days, you are lucky if your home computer has USB connectors for accessories, and plugging in home-made contraptions is definitely not recommended.

Raspberry Pi is different because its design encourages the owner to plug in extra devices and even experiment with your own prototype circuits. Raspberry Pi has standard USB and Ethernet ports, a camera connector, WiFi, and HDMI video output. But the crowning glory of connectivity is its GPIO (general-purpose input/output) 40-pin header, which provides the tinkerer with access to the inner workings of Raspberry Pi. You can even write code to control electronics that are plugged in to the GPIO pins.

What You'll Need

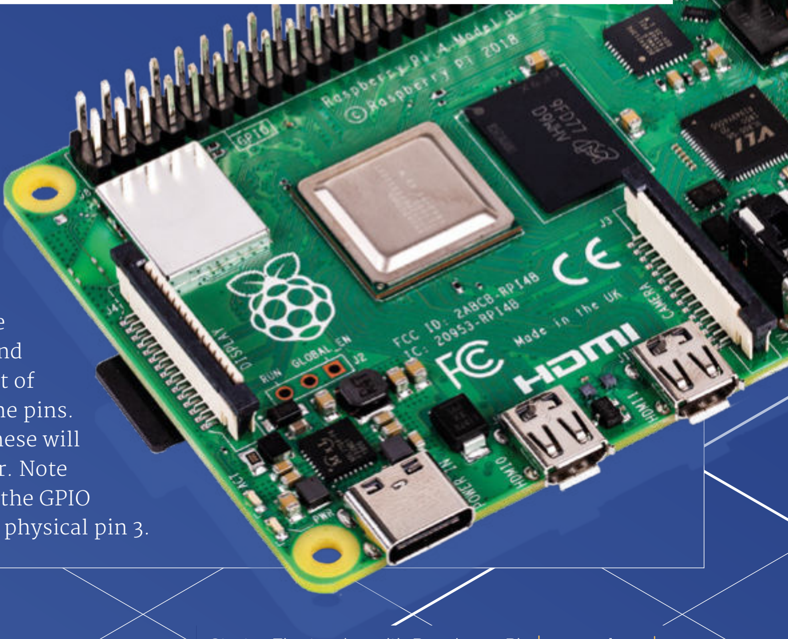
If you are lucky enough to have a local electronics or hobby store, you will no doubt be able to find all the electronic components we will talk about for a few pounds, dollars, or euros. Failing that, there are many fully stocked online outlets that can provide them. All you will need to start with is a breadboard, some jumper leads, a resistor (200 Ω to 470 Ω), and an LED.

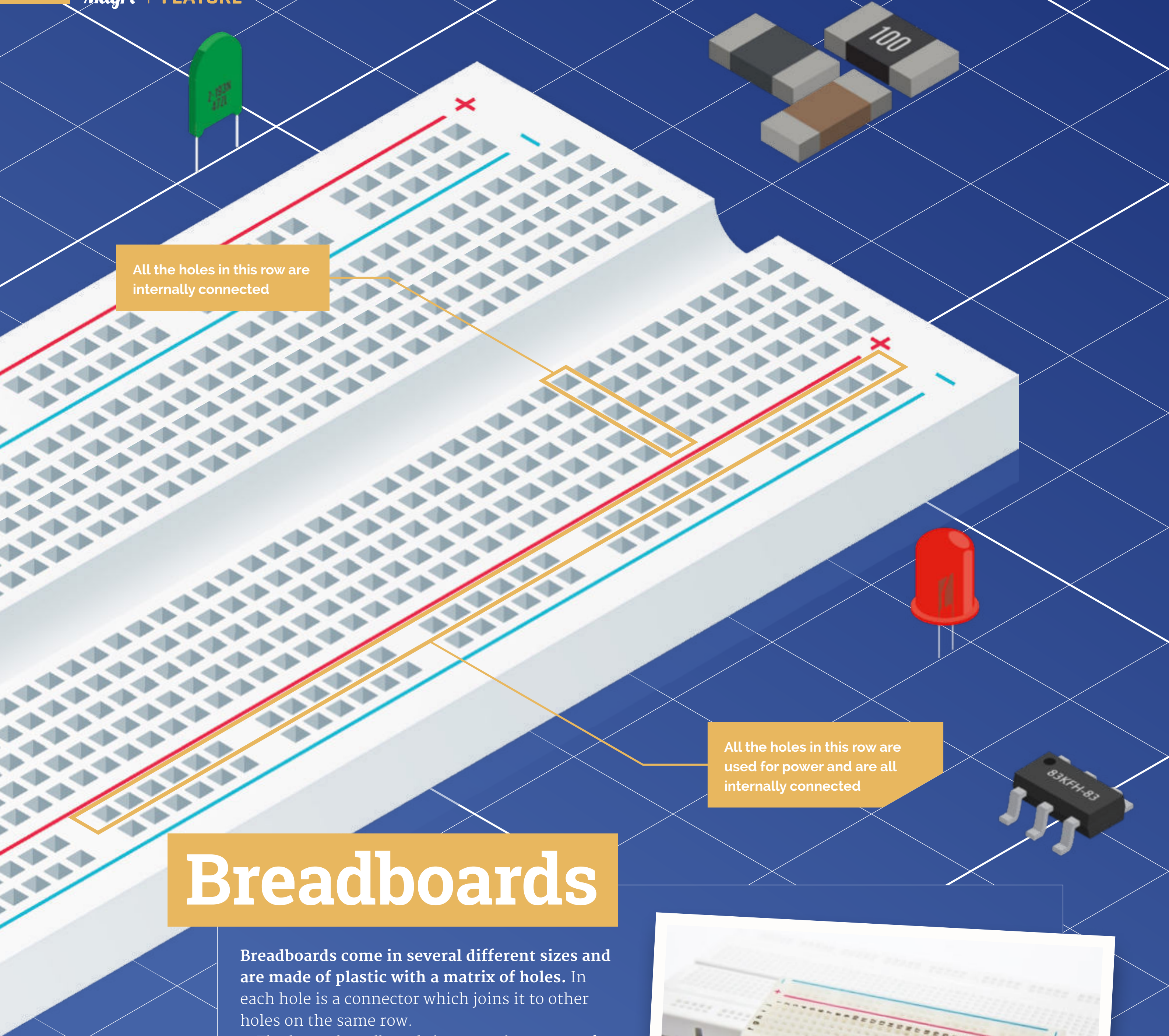


3V3 power	1	2	5V power
GPIO 2 (SDA)	3	4	5V power
GPIO 3 (SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (TXD)
Ground	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18 (PCM_CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3V3 power	17	18	GPIO 24
GPIO 10 (MOSI)	19	20	Ground
GPIO 9 (MISO)	21	22	GPIO 25
GPIO 11 (SCLK)	23	24	GPIO 8 (CEO)
Ground	25	26	GPIO 7 (CE1)
GPIO 0 (ID_SD)	27	28	GPIO 1 (ID_SC)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM_FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM_DIN)
Ground	39	40	GPIO 21 (PCM_DOUT)

GPIO pins

You can use Raspberry Pi's GPIO pins to pass electrical voltage to and from other components. Some of the pins have specific uses, such as power or ground connections, and others can be configured with software programs to send signals between Raspberry Pi and the rest of the circuit. Above is a breakdown of all the pins. Look carefully at physical pins 1 and 6: these will be the pins we use in our LED circuit later. Note that the pin numbers aren't the same as the GPIO numbers; for example, GPIO 2 is actually physical pin 3.




All the holes in this row are internally connected
All the holes in this row are used for power and are all internally connected

Breadboards

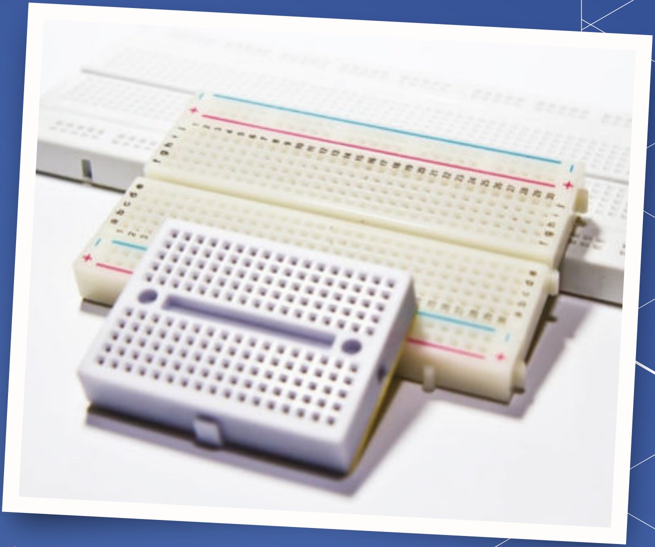
Breadboards come in several different sizes and are made of plastic with a matrix of holes. In each hole is a connector which joins it to other holes on the same row.

The larger breadboards have two long rows of holes at the top and bottom. Sometimes called 'rails', these are generally used for power and ground connections. They are connected together as indicated on the diagram above.

The inner part of the breadboard has two sections of holes. Each section has rows of five holes which are connected as indicated on the diagram. The two halves of the breadboard are a mirror image of each other.

The break along the centre of the breadboard, often known as the 'gutter', enables mounting of integrated circuits (chips) with the legs of the chip going either side of the break.

See **magpi.cc/breadboard** for more info.


▲ Breadboards come in several sizes to enable prototyping of a wide variety of projects

Make an LED light circuit

Let's dive into some practical circuit making with an example of how we can use the GPIO pins on the Raspberry Pi to make a circuit. We will go into the details of what the components are in a little while, but let's look at what a circuit actually looks like on a diagram. We will use the power source of Raspberry Pi to light an LED (light-emitting diode). When electricity is passed through an LED, it glows. We will also need a resistor in our circuit to limit the amount of electricity that goes through the LED. If too much electricity flows, the LED may burn out.

WARNING: When connecting components to the Raspberry Pi, it is always best to have the power off and unplugged in case you make a mistake with your wiring.

01 Turn off the power

Make sure you have shut down your Raspberry Pi and unplugged its power cable. Plugging in jumper leads while switched on may be tolerated by Raspberry Pi, but in some cases it can cause a system crash or even damage to the hardware.

02 Build the circuit

Connect the components as shown in the wiring diagram below. Take careful note of the GPIO pins that you are connecting. The precise positioning of components on the breadboard is not so important, but make sure that the LED's longer (positive) leg is connected to the 3V3 power pin via the jumper lead.

03 Light it up

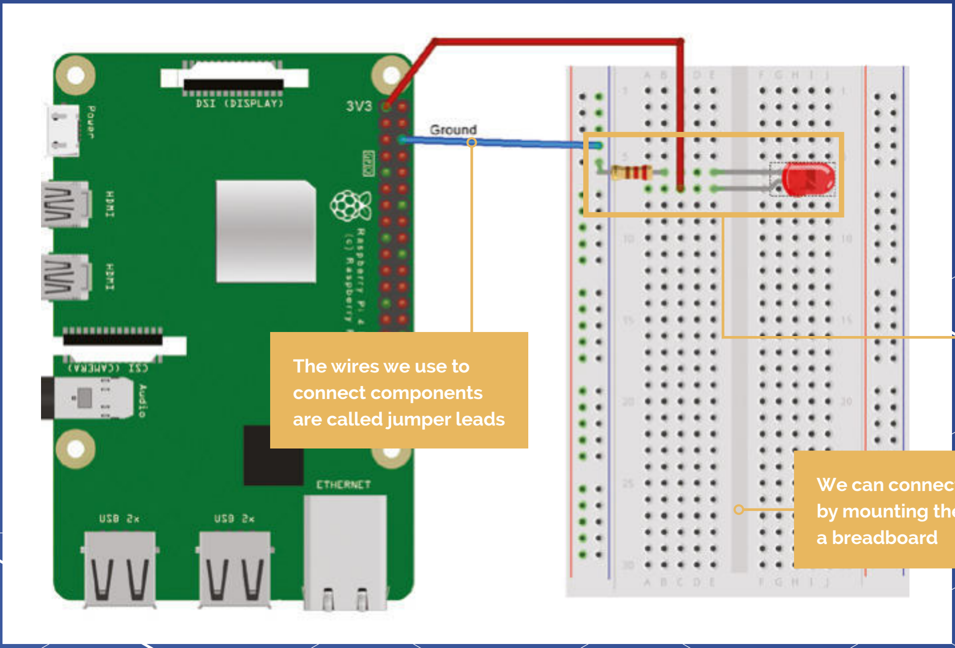
When you are sure that you have connected everything correctly, plug in your Raspberry Pi. After a short time, the LED should light up as soon as the power GPIO pins are energised. If the LED does not light up, try connecting it the other way around, as it will only work one way.

You'll need

To build this simple LED circuit you need the following:

- Breadboard
- LED
- 470 Ω resistor
- 2 × male-to-female jumper wires

Everything can be found in the CamJam EduKit #1 (£5)
magpi.cc/edukit



The wires we use to connect components are called jumper leads

When the resistor and LED are placed in the circuit correctly, the LED will light up

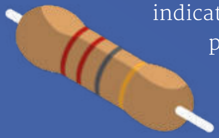
We can connect components by mounting them on a breadboard

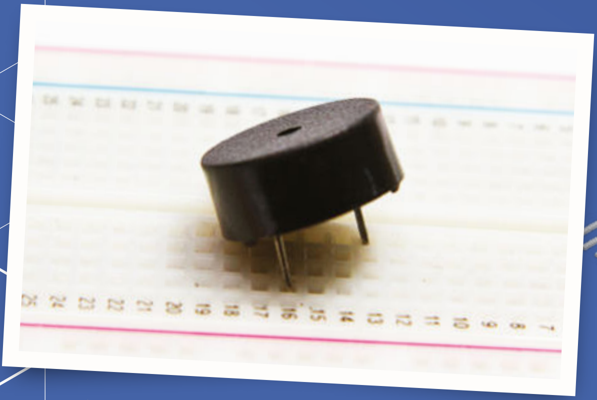
Basic prototyping components

Jumper leads
These connect your components to the Raspberry Pi. They can either have male or female end connectors (you need female for the GPIO pins) and it's a good idea to have a mix of both.

LED

Light-emitting diodes come in various colours. They glow when electricity flows through them. Because they are a diode, the current will only flow in one direction, so make sure that it is connected the right way round: the longer leg is the positive one and should be connected to the power (or GPIO output) pin. Always use a resistor in series with LEDs, otherwise they are likely to draw more current than they can handle and burn out.

Resistor
Resistors are available in various formats, but this is the most common format for prototyping. The coloured rings around the resistor indicate how much resistance it provides. For more details about how to read resistor values, see **magpi.cc/64**.


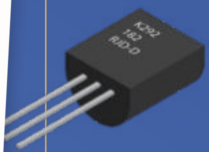

A small piezoelectric speaker is good for making simple noises

Switch
Switches make and break circuits. When connected in series in a circuit, a switch will allow or stop the flow of electricity. You can get switches that stay in the position in which they are set, or ones that you need to keep your finger on to keep the circuit connected.

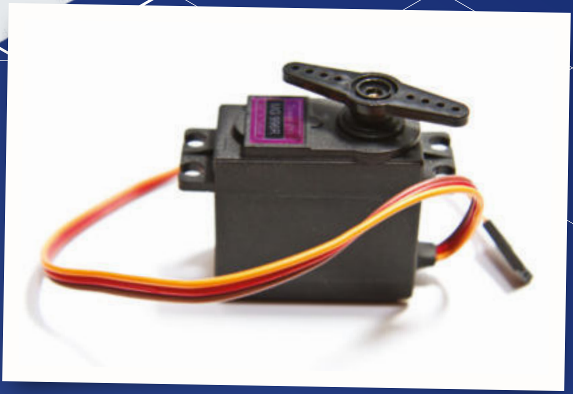
Speaker
Speakers can be huge and very loud or they can be very small – small enough to fit on a breadboard. These small speakers are called piezoelectric and are great for making robotic noises, but not much good for hi-fi music applications.

Capacitor

Capacitors store and release electrical charge. They can be used to even out the electrical current or provide a greater flow for a short time. Be careful when using capacitors, as they can still hold charge after the circuit has been disconnected.

Transistor
Transistors are used in many types of circuits. They can be used either as an amplifier, which means that a small current goes in but a bigger current comes out, or as an electrical switch to change the amount of electricity flowing through circuit. A transistor has three legs or connectors: the emitter, the collector, and the base.


Servo
Servos are used a lot in robotics for making things move. They can be connected directly to Raspberry Pi via the GPIO pins, but there are also



A servo is a special type of motor that allows for precise control of its rotational (or linear) position

extra plug-in boards (or HATs) that make it easier to connect many servos and control them more easily. Normally we use digital servos which use PWM (pulse-width modulation) signals to turn the armature on the servo from one position to another.

Batteries

Batteries can be all kinds of shapes and sizes, but for simple electronics projects you will normally be using 1.5 V AA or AAA type cylindrical batteries or 9 V square ones. Make sure you always have some spares or perhaps you may want to use rechargeable ones. Rechargeable batteries cost more, but can be used over and over. Make sure you connect batteries correctly and never directly connect the two contacts together, as even small voltage batteries can get very hot or even explode if 'short-circuited'.

Potentiometer

Potentiometers are variable resistors and usually have a twisting knob or a slider to change the resistance. They have three connectors and can be used as game controllers or volume controls. Be careful if you are using a potentiometer instead of a resistor in your LED circuit: if you turn the resistance right down to zero, you may burn out your LED.

A potentiometer produces an analogue output that may be converted to digital by an ADC

Prototyping using kits

CamJam EduKits

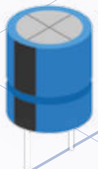
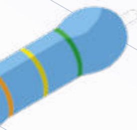
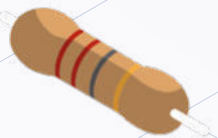
There are currently three CamJam EduKits available: a starter kit with all the components and more to make our LED circuit, a sensors kit to explore getting input from the world around you, and a robotics kit with wheels and motors to make your Raspberry Pi into a robot.
magpi.cc/camjamkits

Adafruit Parts Pal

The Adafruit Parts Pal is a more comprehensive kit. It contains many popular components and prototyping parts, including LEDs, resistors, cables, sensors, and mechanical parts.
magpi.cc/partspal

Elecrow CrowPi Educational Kit

This is the daddy of all-in-one experimentation kits. There are chips and sensors, dials, speakers, numeric displays, and even a 7-inch HDMI touchscreen. It was developed with a Kickstarter appeal and is available directly from Elecrow in the US, or Amazon in the UK.
magpi.cc/crowpi

Getting started with electronics: LEDs and switches

MAKER

Simon Monk

Simon divides his time between writing and designing products for MonkMakes Ltd. Some of his better-known books include *Programming Raspberry Pi* (TAB) and *The Raspberry Pi Cookbook* (O'Reilly).

simonmonk.org

Use Raspberry Pi's GPIO pins to buid two simple electronic projects. By **Simon Monk**

In this tutorial, you will learn how to make two simple projects that use LEDs and push-button switches that are controlled by a Python program running on your Raspberry Pi. The first project uses an RGB (red, green, blue) LED to interface to the popular Cheerlights project. Cheerlights (**cheerlights.com**) allows users all over the internet to set each other's LEDs to different colours just by tweeting.

The second project is a reaction timer using LEDs and push-buttons to test the speed of your reactions.

You'll Need

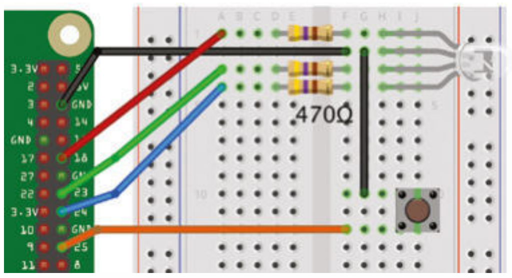
- The Mu Python editor **codewith.mu**
- Solderless breadboard
- 5 × female-to-male jumper wires
- Male-to-male jumper wire
- 2 × red LEDs
- RGB common cathode LED
- 3 × 470 Ω resistors
- 2 × tactile push-buttons

These components are all included in a MonkMakes kit: **magpi.cc/pibox1**

Set up some Cheerlights

01 Install the code

Before fetching the code from the internet, you should run Mu, which you will find in the Programming section of your main menu. If it's not there, update your system to the latest version of Raspbian (**magpi.cc/raspbianupdate**).



▲ **Figure 1** The Cheerlights wiring diagram

Running Mu ensures that the **mu_code** directory is created, into which we will now copy the program code. To do this, open a Terminal window and run the commands:

```
wget http://monkmakes.com/downloads/pb1.sh
sh pb1.sh
```

This will copy the programs used in this tutorial into the **mu_code** directory, along with some other programs.

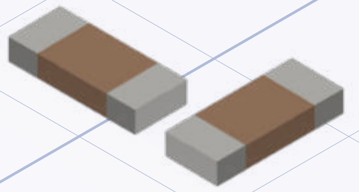
" Push the component legs into the breadboard at the positions shown "

02 Place components onto breadboard

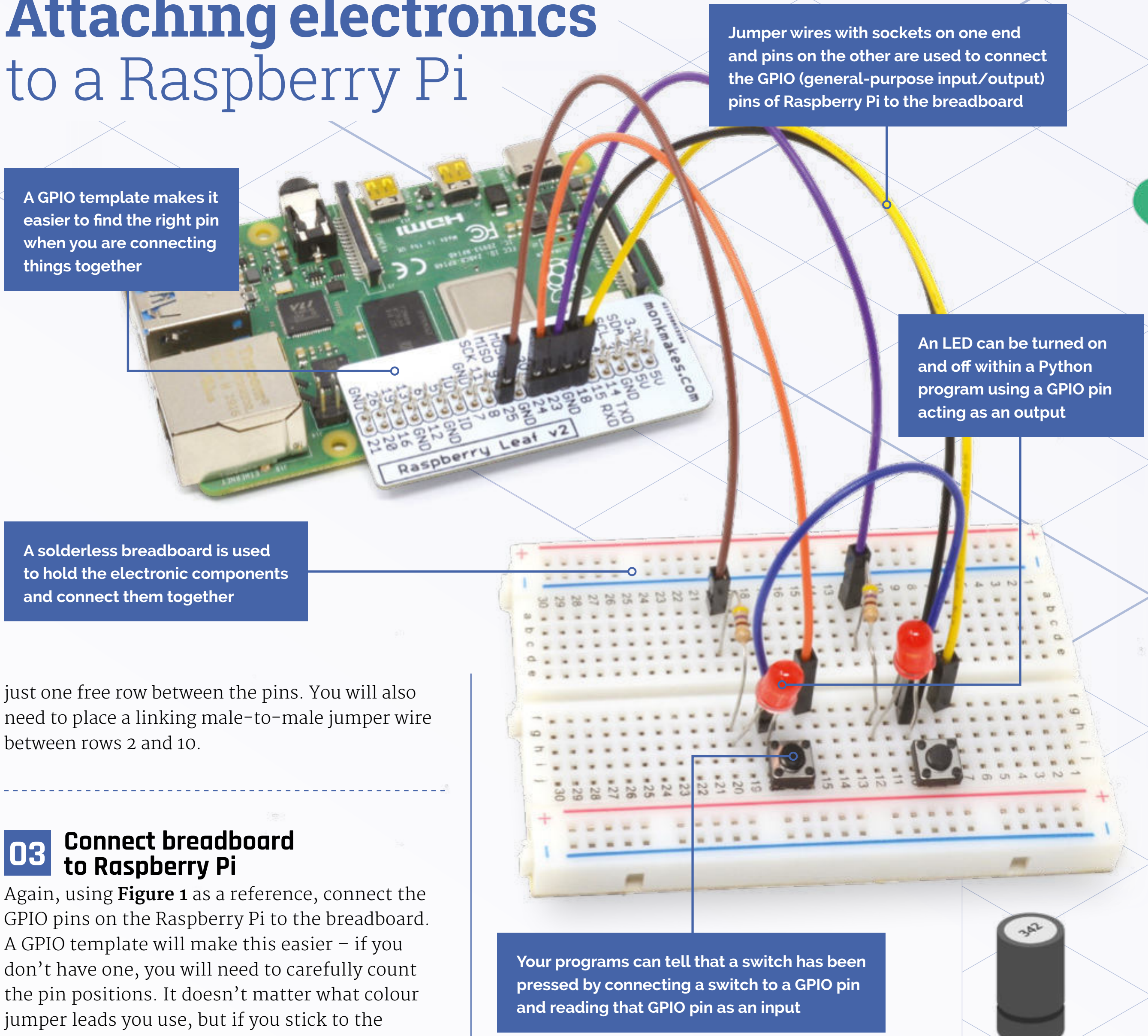
Using **Figure 1** as a reference, push the component legs into the breadboard at the positions shown. Bend the resistor legs so that they fit into the holes.

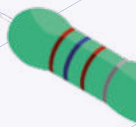
Each hole in a row of five holes on the breadboard is connected together under the plastic. So, its very important to get the right row for your component leg.

The resistors can go either way around, but the RGB LED must go the right way around, with its longest leg to row 2 (the one without a resistor). The push-button used in the MonkMakes kit has just two legs, but many similar buttons have four legs. If you have a four-legged version, put it on the breadboard in the orientation that leaves



Attaching electronics to a Raspberry Pi


Jumper wires with sockets on one end and pins on the other are used to connect the GPIO (general-purpose input/output) pins of Raspberry Pi to the breadboard
A GPIO template makes it easier to find the right pin when you are connecting things together
An LED can be turned on and off within a Python program using a GPIO pin acting as an output
A solderless breadboard is used to hold the electronic components and connect them together
Your programs can tell that a switch has been pressed by connecting a switch to a GPIO pin and reading that GPIO pin as an input
monkmakes.com
Raspberry Leaf v2



just one free row between the pins. You will also need to place a linking male-to-male jumper wire between rows 2 and 10.

03 Connect breadboard to Raspberry Pi

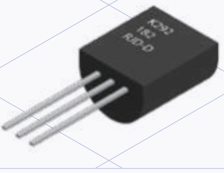
Again, using **Figure 1** as a reference, connect the GPIO pins on the Raspberry Pi to the breadboard. A GPIO template will make this easier – if you don't have one, you will need to carefully count the pin positions. It doesn't matter what colour jumper leads you use, but if you stick to the colours used in the diagram, it's easier to check that your wiring is correct.

04 Running the program

To use this project, your Raspberry Pi must be connected to the internet. Load and run the program **04_cheerlights.py** using Mu. After a few seconds, the LED will automatically set itself to the current Cheerlights colour, checking every ten seconds. Pressing the button will turn the LED off until the Cheerlights colour changes.

05 Tweet a new colour

Now that your Raspberry Pi is looking out for changes to the Cheerlights colour, anyone can simply send a tweet mentioning **@cheerlights** and the name of a colour; your LED should then change to that colour. You can test this out by sending a tweet such as '@cheerlights red' and after a few seconds your LED should change colour. You will find that after a few minutes, the colour probably changes as someone else sets the Cheerlights colour.



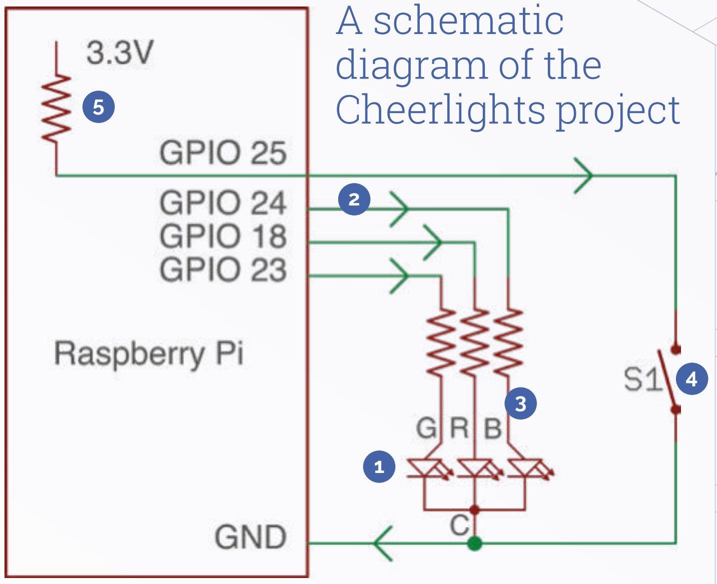
1. The RGB LED is actually three LEDs in one: red, green, and blue. Changing the power going to each LED (controlled by a separate GPIO pin) changes the overall colour.

2. GPIO 24 acts as an output. Current flows out of GPIO 24, through the resistor, through the blue LED and back to Raspberry Pi's GND (ground connection).

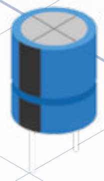
3. An LED will draw as much current as it can, so each LED needs a resistor to reduce the current, protecting the LED and/or the GPIO pin of Raspberry Pi.

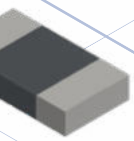
4. When the switch is pressed, it connects GPIO pin 25 (acting as an input) to GND (0V).

5. An internal pull-up resistor keeps GPIO 25 at 3.3V until the switch is pressed – that overrides the effect of the resistor, making GPIO 25 0V. Without this, GPIO 25 would be a floating input liable to false triggering from electrical noise.



A schematic diagram of the Cheerlights project





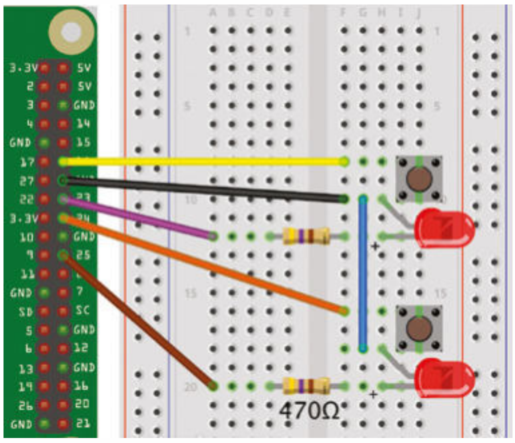
Build a reaction timer

01 Dismantle the breadboard

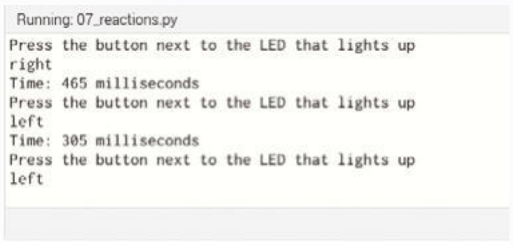
First, pull the jumper leads off the GPIO pins on the Raspberry Pi and then pull all the components and wires off the breadboard so that it is ready for the next project.

02 Place the components

This time, using **Figure 2** as a guide, push all the component legs into the breadboard at the positions shown. It doesn't matter which way round the resistors and buttons go, but the LEDs have a positive and negative end, so must go the correct way around. The positive end of the LED (marked '+' on the diagram) is the longer leg and this should go to the same row on the breadboard as the resistor.



▲ **Figure 2** The Reaction Timer wiring diagram



Running: 07_reactions.py

```
Press the button next to the LED that lights up
right
Time: 465 milliseconds
Press the button next to the LED that lights up
left
Time: 305 milliseconds
Press the button next to the LED that lights up
left
```

▲ **Figure 3** The Reaction Timer results in the Mu console

03 Connect breadboard to Raspberry Pi

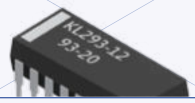
Using **Figure 2** as a reference, connect the GPIO pins on the Raspberry Pi to the breadboard using five female-to-male jumper wires.

04 Running the program

To use the reaction timer, load and run the program **07_reactions.py** in Mu. When the program starts, you will notice that the bottom part of the Mu window shows a message telling you to 'Press the button next to the LED that lights up' (**Figure 3**).

After a random amount of time, one of the LEDs will light, and you should press the button next to that LED as quickly as possible. You will then get a message telling you how many milliseconds you took to press the button.

The code includes checks to make sure you don't try to cheat by pressing both buttons at once, or pressing the buttons before an LED has lit.



04_cheerlights.py

▸ Language: **Python 3**

DOWNLOAD THE FULL CODE:

magpi.cc/pibox1git

```python
# 04_cheerlights.py
# From the code for the Box 1 kit for the Raspberry
Pi by MonkMakes.com

from gpiozero import Button, RGBLED
from colorzero import Color
import time, requests

update_period = 10 # seconds
led = RGBLED(red=18, green=23, blue=24)
button = Button(25)

cheerlights_url = "http://api.thingspeak.com/
channels/1417/field/2/last.txt"
old_color = None

def pressed():
    led.color = Color(0, 0, 0)  # LED off
button.when_pressed = pressed

while True:
    try:
        cheerlights = requests.get(cheerlights_url)
        color = cheerlights.content # the color as text
        if color != old_color:
            led.color = Color(color) # the color as an object
            old_color = color
    except Exception as e:
        print(e)
    time.sleep(update_period) # don't flood the web service
```

07_reactions.py

▸ Language: **Python 3**

```python
# 07_reactions.py
# From the code for the Box 1 kit for the Raspberry Pi by MonkMakes.com

from gpiozero import LED, Button
import time, random

left_led = LED(25)
right_led = LED(23)
left_switch = Button(24)
right_switch = Button(18)

# find which buttons pressed 0 means neither, -1=both, 2=right, 1=left
def key_pressed():
    # if button is pressed is_pressed will report false for that input
    if left_switch.is_pressed and right_switch.is_pressed:
        return -1
    if not left_switch.is_pressed and not right_switch.is_pressed:
        return 0
    if not right_switch.is_pressed and left_switch.is_pressed:
        return 1
    if right_switch.is_pressed and not left_switch.is_pressed:
        return 2

while True:
    left_led.off()
    right_led.off()
    print("Press the button next to the LED that lights up")
    delay = random.randint(3, 7) # random delay of 3 to 7 seconds
    led = random.randint(1, 2) # random led left=1, right=2
    time.sleep(delay)
    if (color == 1):
        print("left")
        left_led.on()
    else:
        print("right")
        right_led.on()
    t1 = time.time()
    while not key_pressed():
        pass
    t2 = time.time()
    if key_pressed() != led : # check the correct button was pressed
        print("WRONG BUTTON")
    else:
        # display the response time
        print("Time: " + str(int((t2 - t1) * 1000)) + " milliseconds")
```

NexDock 2 is a laptop dock that acts as a keyboard, mouse, and screen for Raspberry Pi (and Android phones)

NexDock 2

▶ NexDock ▶ **nexdock.com** ▶ £226 / $259

This laptop dock could be just the portable Raspberry Pi solution we've been looking for. By **Lucy Hattersley**

SPECS

DIMENSIONS: 317×215×15.9 mm

WEIGHT: 1420 g

DISPLAY: 13.3-inch IPS screen, 1920×1080 FHD resolution, 16:9 aspect ratio

BATTERY: 51 Wh, 7.6 V, 6800 mAh

INPUT PORTS: 1 × USB-C 3.1 with DisplayPort, 1 × HDMI-in (1.4a) port

PORTS: 1 × USB-C PD Charging Port, 1 × USB-C 3.0, 1 × USB-A 3.0, 1 × 3.5 mm audio, 1 × microSDXC reader

AUDIO: 4 × 1 W speakers

NexDock 2 is a laptop dock, or 'lapdock' as some folks have taken to calling them. These devices are few and far between, but compatible ones have considerable charm and value for Raspberry Pi users.

Laptop docks are primarily designed to extend Android phones into working laptops. However, plug your Raspberry Pi 4 into NexDock 2 and you get a fully-functioning Raspberry Pi laptop. Raspberry Pi provides the brains; NexDock 2 is the keyboard, mouse, and screen.

And what a lovely screen it is: the 13.3-inch IPS 1920×1080 display is a delight to look at, even if the chunky borders are a bit retro. The edge-to-edge full-sized backlit keyboard is equally impressive, with responsive chiclet-style keys that are suitably clicky. Typing is a breeze.

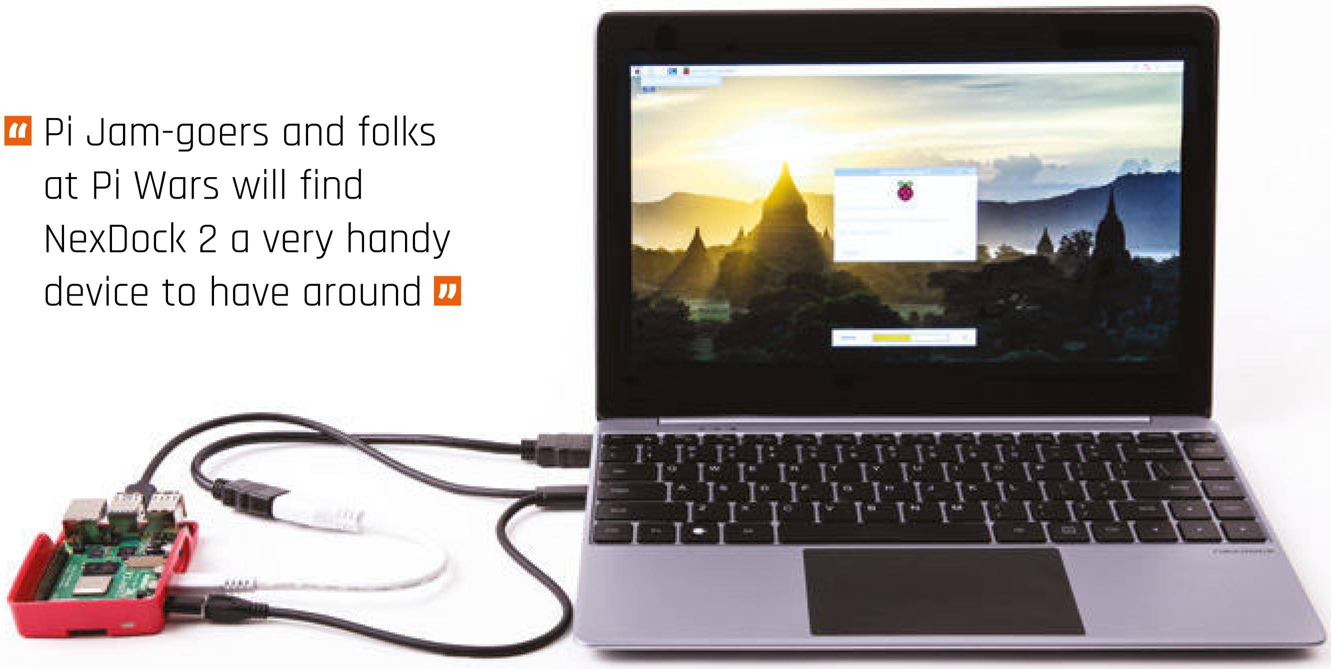
The whole thing is set off in a grey and black style that's far more professional than its price tag warrants.

And we're trying hard to ignore the unfortunate bulge in the plastic hinge above our **F2** key. Perhaps we got an early run that slid through the checks.

Battery life

Inside NexDock 2 sits a 51 Wh battery that provides power to the screen and keyboard and Raspberry

> Pi Jam-goers and folks at Pi Wars will find NexDock 2 a very handy device to have around



Pi. We ran our fully charged unit with Raspberry Pi 4 playing YouTube videos from 10:10 to 15:22, just over five hours of solid playback.

A supplied 60 W USB-C adapter is used for charging.

Plugging in

In the box is an HDMI cable with a micro-HDMI to HDMI adapter for Raspberry Pi 4.

Meanwhile, a USB-C cable splits out to USB-A and micro-USB (for power and keyboard/mouse connection). That's a lot of cabling compared to a regular laptop. The nest of cables is an issue.

We managed to bend the connection on the HDMI-to-micro-HDMI adapter, and replaced it with the white extension cable (pictured above). Be sure to disconnect everything when you pack up.

Achilles' trackpad

The weakest link is the trackpad. There's no thumb rejection – so as you type, the cursor jumps around the screen. And if you use a thumb to click, the cursor jumps from the clicked point.

Our solution is to attach a USB mouse and disable the trackpad with xinput:

```
  xinput --set-prop "SINO WEALTH USB KEYBOARD
Mouse" "Device Enabled" 0
```

To re-enable the trackpad:

```
  xinput --set-prop "SINO WEALTH USB KEYBOARD
Mouse" "Device Enabled" 1
```

The lack of hardware and software integration soon becomes apparent. Closing the screen shuts off the power to the display, but not immediately to Raspberry Pi, which carries on running for a short time until the power is yanked. We took to using `shutdown -h now` when done and removing all the cables.

Digital nomad

Using Raspberry Pi with NexDock 2 as your go-to laptop is stretching credibility. The finicky trackpad, nest of wires and dongles, and lack of creature comforts will drive you round the bend.

However, NexDock 2 is ideal for users who need to power up a Raspberry Pi at an event. We imagine Pi Jam-goers and folks at Pi Wars will find NexDock 2 a very handy device to have around.

▲ There is a lot of cabling involved when using NexDock 2 with Raspberry Pi as a laptop

Verdict

An incredibly useful, yet far from perfect, solution to using Raspberry Pi on the move. Despite its flaws, it is, by far, the best solution we have found to this particularly thorny problem.

8/10

RedBoard+

▶ Red Robotics ▶ **redrobotics.co.uk** ▶ £31 / $40

A feature-packed robotics board with the ability to drive powerful motors. By **Phil King**

SPECS

MOTOR CONTROL: 2 × 6 amp H-bridges with fully proportional speed control

CONNECTIONS: 13 × GPIO (plus power and ground), 4 × I²C, 3 × analogue inputs, 10-pin useful header

POWER: 7V – 24V battery input, with reverse polarity protection

FEATURES: Four-channel ADC, four-channel 5V level shifter, power switch, programmable button, optional daughterboard, and OLED

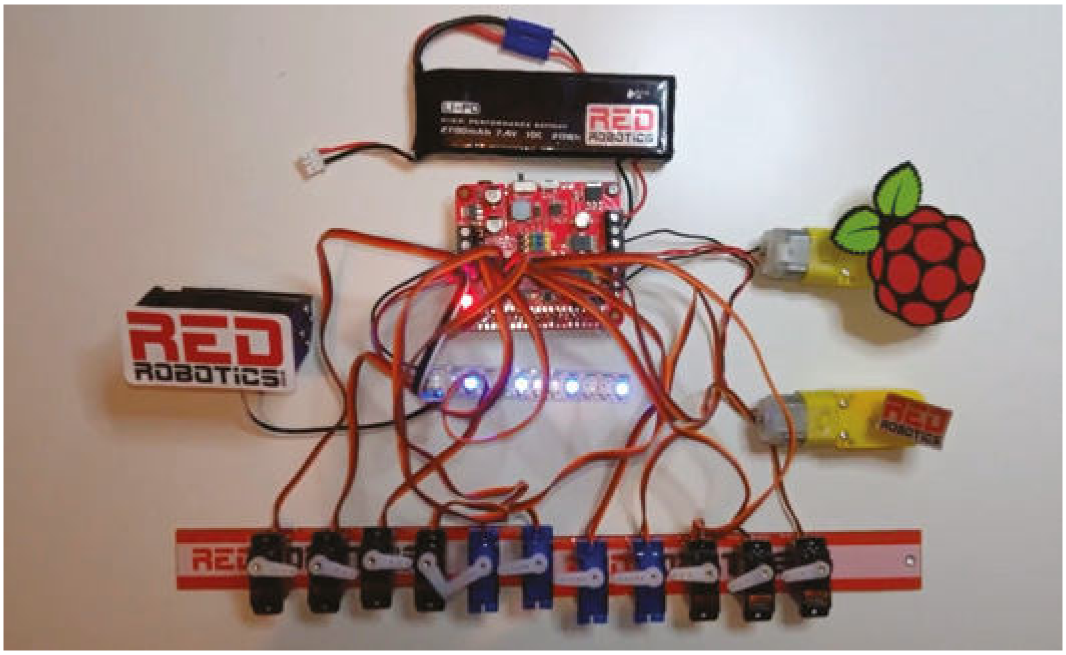
While there's no shortage of robotics controller boards available for Raspberry Pi, this one has two main selling points. Firstly, it boasts two really powerful motor drivers. Secondly, an astonishing number of features are crammed onto this full-size board – it seems its designer (Neil Lambeth of Red Robotics) has thought of everything you might need for a robotics project.

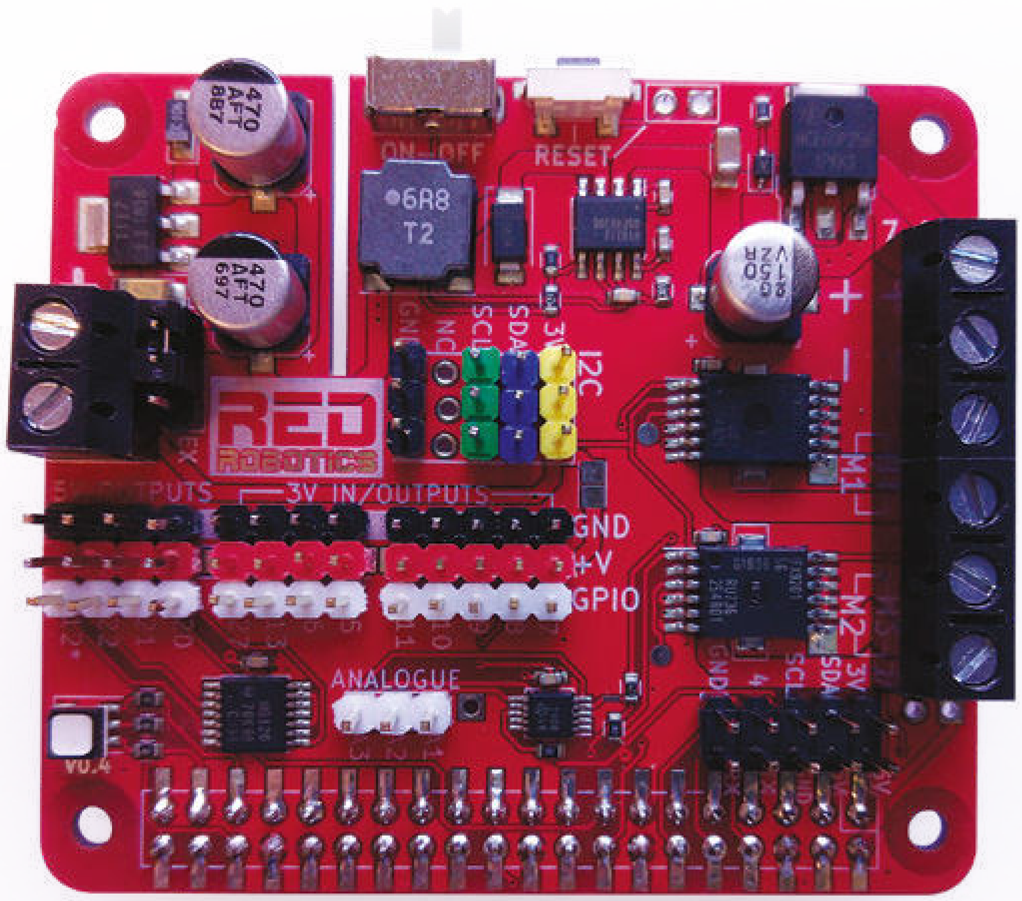
Serious power

Two on-board H-bridges provide up to 6A of continuous current per channel to two motors, or two sets of motors, with fully proportional speed control. While there's no facility to independently control four motors, as on the ZeroBorg for example, this won't be a deal-breaker for most robot builders.

"The input voltage for external power is 7V to 24V, so you have a wide range of options"

Motors are connected via screw terminals, as is your external power source. The input voltage for this is 7V to 24V, so you have a wide range of options, from AA battery packs to LiPo cells – LiPo is recommended for driving big motors. For a good example of the kind of power it can deliver, with the right motors, check out Neil's test-drive of his four-wheeled robot zipping around a lawn at great speed: **magpi.cc/redboardtest**.


▶ A demo showcasing the RedBoard+'s multifunctionality with twelve servos, NeoPixels, and motors connected



A cornucopia of connectivity, the RedBoard+ has multiple breakout headers, including 5V outputs and analogue inputs

The external battery also powers your Raspberry Pi, RedBoard+'s on-board BEC delivering a steady 5.2V at 3A, which is enough for a Raspberry Pi 4.

One nice touch is the inclusion of a power switch to turn the battery power on and off. There's also a user-programmable push-button: by default it's set to reboot Raspberry Pi with a medium press, and shut it down with a longer press. A short press will cause an on-board RGB LED to flash repeatedly in red, green, blue to show the last number in the IP address – handy for SSHing in.

Servo central

The RedBoard+ offers an impressive range of connectivity options. The main header breaks out 13 GPIO pins (including SPI, GPIO 7–11), with power and ground for each, so you can connect numerous servos and/or other electronics. Making use of an on-board voltage level shifter, four of the sets of pins offer 5V power, while the GPIO 12 pin can be used to control NeoPixels with PWM.

If you want to use larger 7.4V robot servos, there's an option to connect a separate power supply via two screw terminals, although this does set all the main header power pins to 7.4V.

The RedBoard+ also features a three-pin header for analogue inputs (up to 3.3V) using the on-board TI ADS101x four-channel ADC – its other channel is dedicated to LiPo battery voltage monitoring.

That's not all! The board includes a header for four I²C channels, plus a ten-pin 'useful header' which breaks out I²C, Rx/Tx, 3.3V/5V power, and ground pins – this can be used with an optional daughterboard and mini OLED to show battery stats, IP address, etc.

Software

Controlling everything is made easy using the Python software library – if you don't want to install it manually, there's even a preconfigured Raspbian image available. Plentiful code examples – including controlling various functions with a wireless gamepad – should help you get started, along with the detailed instructions on GitHub.

While RedBoard+ is probably overkill if you just want to drive a couple of standard low-power motors, it's an amazing feature-packed board. To get an idea of what it can really do, check out Neil's demo video showcasing the multifunctionality: **magpi.cc/redboarddemo**.

Verdict

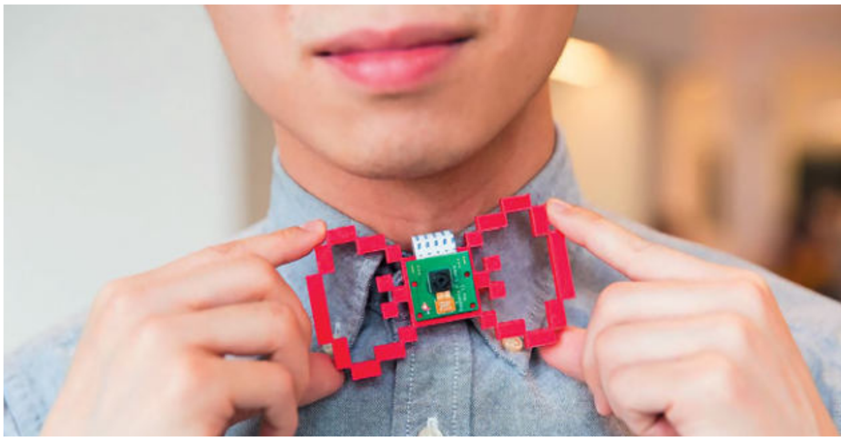
With its powerful motor drivers and huge range of connections for servos and electronics, we can see this becoming a favourite board used in events like Pi Wars.

10/10

10 Best: Raspberry Pi wearable projects

Upgrade your day-to-day life with these amazing projects that you can wear

Commercial wearable computing has come a long way in the last decade. Google Glass was a thing, virtual reality headsets are still a thing, while watches can be used just as they were in *Dick Tracy* and *Power Rangers*. With the help of a Raspberry Pi, you can go far beyond commercially available products and make your own amazing things.



▲ 3D-printed **spy cam bowtie**

Fun yet conspicuous

This very cute project won't exactly get you any spy gigs, but it does mean your spying is a little more consensual with such a visible and cool-looking project.

magpi.cc/bowtie

▲ Windows 98 **Wrist Watch**

Impressively classic

Is having Windows 98 on your wrist useful? Probably not, but being able to build a device that allows it is very cool and quite funny.

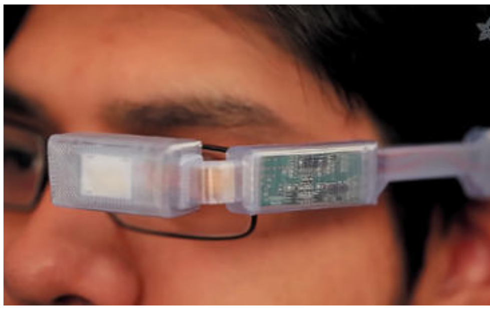
magpi.cc/w98watch

▲ **Iron Man** Arc Reactor

No scraps required

Want to make your Iron Man suit really stand out, without spending your life savings on a fully articulated suit? Check out how to make an Arc Reactor.

magpi.cc/arcreactor

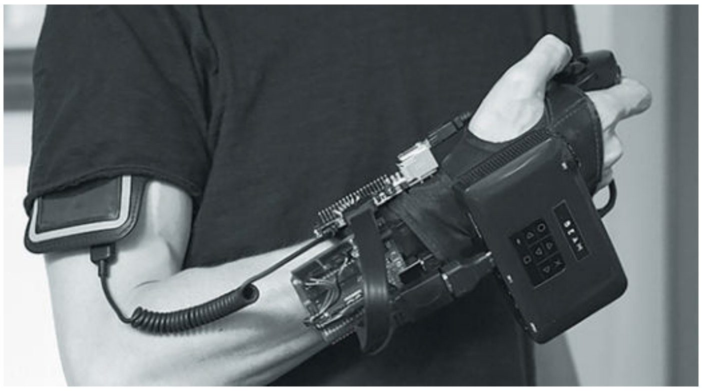


▶ **DIY** Glass

Augment your reality

While perhaps a little more bulky than the old Google Glass, this DIY version is much cheaper and you can still actually make/get one!

magpi.cc/diyglass



▲ VIDEOBLAST_R

Handheld projector

Designed to give the user ultimate artistic freedom, it features projection mapping tech and can hold up to eight audiovisual pieces that you can project whenever you wish.

magpi.cc/videoblastr

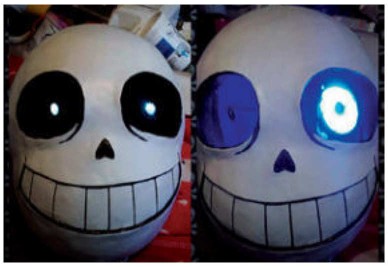
▸ Raspberry Pi **Pip-Boy 3000**

Apocalyptic wearable

Recreate the famous wrist computer from the Fallout game series – it's a pretty popular and amazing project!

magpi.cc/pipboy3000

▾ **NeoPixel** cosplay eyes



Light-up costuming

Using NeoPixels with a Raspberry Pi is simpler than some folks make it out to be. Try out the easy way with our tutorial, and soup up your cosplay.

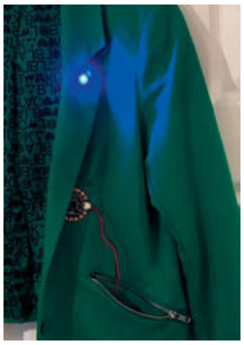
magpi.cc/neopixeleyes

▾ **Wearable** Pi Zero Camera

Low-key body camera

This build is pretty simple but very effective, and truly shows off the power of a Raspberry Pi Zero and just how much it can do at its tiny size.

magpi.cc/zerocam



◂ Raspberry Pi **jacket**

Wearable-optimised LEDs

This project uses FLORA, a sewable bit of IoT tech that can interact with a Raspberry Pi. Learn how to control it and add it to a jacket or coat!

magpi.cc/pijacket

▾ Social Media **without the Internet**

Wearable art

This amazing art project asks people to act as if they're on social media in the real world. It's also an incredible piece of engineering.

tuangstudio.com

Learn SQL and database design with Raspberry Pi

SQL and NoSQL databases a mystery? **PJ Evans** has some recommendations to get you started

SQLite

CREATOR

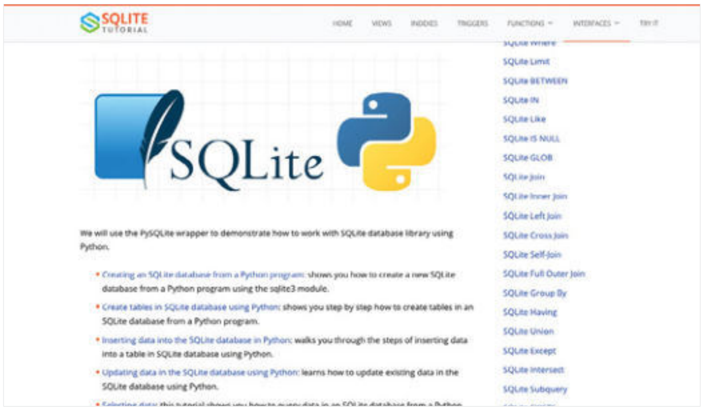
SQLite Tutorial

Price:
FREE

sqlitetutorial.net

SQLite is the perfect database for learning SQL (Structured Query Language, used to get data in and out of a database). It is unique as it does not rely on a running server: instead, the database is stored in a regular file. So, you can avoid all the sysadmin stuff and get straight into writing SQL. Like all the database platforms covered here, it is open-source and a breeze to install on Raspbian.

Despite its diminutive nature, SQLite is fast and powerful, making it suitable for smaller projects. SQLite's site is basic, but **sqlitetutorial.net** provides free interactive training. As well as covering SQL for SQLite (all platforms have slightly different implementations of SQL), there is a Python-specific course at **magpi.cc/sqlitepython**. Here, you can learn how to use PySQLite, the Python bindings for SQLite. Not only is the SQL language covered, but how to create databases and tables in Python. This makes for a comprehensive combination of courses.



We will use the PySQLite wrapper to demonstrate how to work with SQLite database library using Python.

- Creating an SQLite database from a Python program: shows you how to create a new SQLite database from a Python program using the sqlite3 module.
- Create tables in SQLite database using Python: shows you step by step how to create tables in an SQLite database from a Python program.
- Inserting data into the SQLite database in Python: walks you through the steps of inserting data into a table in SQLite database using Python.
- Updating data in the SQLite database using Python: learns how to update existing data in the SQLite database using Python.
- Selecting data: this tutorial shows you how to query data in an SQLite database from a Python

Great SQL books

Prefer learning though the printed word? Try these

THE DEFINITIVE GUIDE TO SQLITE

Grant Allen and Michael Owen's book on SQLite is a comprehensive look at this 'micro' database. Despite its small footprint, the authors reveal an incredibly powerful engine and some surprising applications including embedded systems.
magpi.cc/sqllitedef

MYSQL FOR PYTHON

Python being the preferred choice of language of the Raspberry Pi Foundation, this book is a perfect introduction to using Python's MySQL wrappers. The book is formatted as a series of tutorials that increase with complexity.
magpi.cc/mysqlpython

MONGODB AND PYTHON

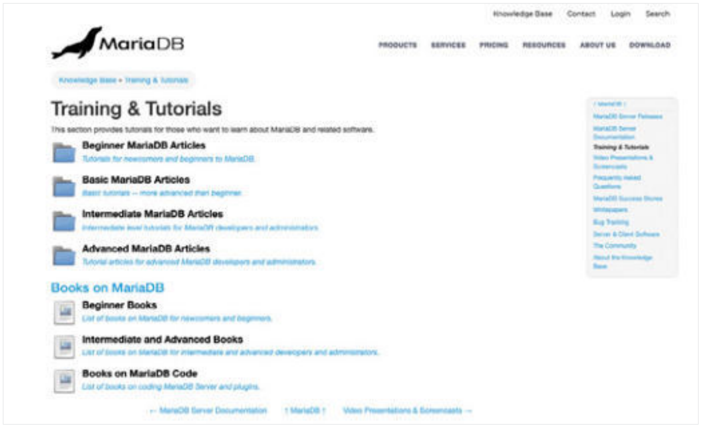
Like the aforementioned *MySQL for Python* book, this one gets right to the point. It uses the 'recipe' format, so you can find a close match for the problem you want to solve, and see how using MongoDB can help.
magpi.cc/mongodbpython

MariaDB/MySQL

CREATOR
MariaDB

Price:
Free

mariadb.com


Knowledge Base Contact Login Search

MariaDB

PRODUCTS SERVICES PRICING RESOURCES ABOUT US DOWNLOAD

Knowledge Base » Training & Tutorials

Training & Tutorials

This section provides tutorials for those who want to learn about MariaDB and related software.

Beginner MariaDB Articles
Tutorials for newcomers and beginners to MariaDB.

Basic MariaDB Articles
Basic tutorials — more advanced than beginner.

Intermediate MariaDB Articles
Intermediate level tutorials for MariaDB developers and administrators.

Advanced MariaDB Articles
Tutorial articles for advanced MariaDB developers and administrators.

Books on MariaDB

Beginner Books
List of books on MariaDB for newcomers and beginners.

Intermediate and Advanced Books
List of books on MariaDB for intermediate and advanced developers and administrators.

Books on MariaDB Code
List of books on coding MariaDB Server and plugins.

← MariaDB Server Documentation ↑ MariaDB ↑ Video Presentations & Screencasts →

The next logical step from SQLite is to move to a full relational database server. A popular choice is MariaDB, an open-source project based on, and fully compatible with, MySQL. MySQL is probably one of the most popular database servers of all time. Both variants are free, but MariaDB boasts a new 'engine' that provides significant speed increases. Installing the server is straightforward and it runs surprisingly fast, even on older Raspberry Pi hardware. Once installed, you've more to think about than with SQLite, such as users and permissions. Thankfully, MariaDB offers online courses for free at **magpi.cc/mariadbtraining** – a nicer way to learn than ploughing through dry documentation. This is a powerful platform: it can take tables containing millions of rows in its stride. It's possible to use MariaDB/MySQL for anything from your own logging project, to enterprise-grade applications.

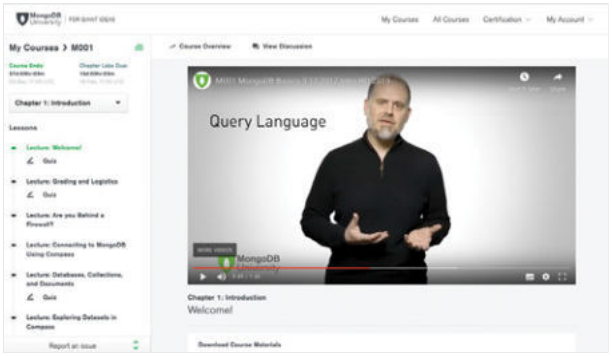
MongoDB

CREATOR
MongoDB

Price:
Free

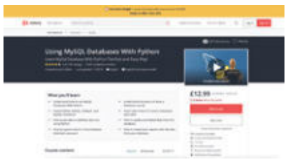
mongodb.com

Certain applications or projects require a different type of database. If you're looking at storing lots and lots of simple data and don't require the kind of data extraction that relational databases offer, then a NoSQL (or document) database may be for you. This family of databases specialise in high-volume, fast data storage and retrieval. They are especially popular when the job is to 'just get the data and store it as quickly as possible'. Rather than using the traditional model of tables, columns, and rows, document databases use schemas and key-pairs to store data. MongoDB is a popular open-source server. It's very easy to install, and you can undertake a full course at **magpi.cc/mongodb**.


My Courses › M001
Chapter 1: Introduction
Lessons
Lecture: Welcome!
Quiz
Lecture: Grading and Logistics
Quiz
Lecture: Are you Behind a Firewall?
Lecture: Connecting to MongoDB Using Compass
Lecture: Databases, Collections, and Documents
Quiz
Lecture: Exploring Datasets in Compass
Course Overview
View Discussion
Query Language
Chapter 1: Introduction
Welcome!
Download Course Materials

Udemy courses

Learn online and interactively with these courses



INTRO TO SQLITE DATABASES FOR PYTHON PROGRAMMING

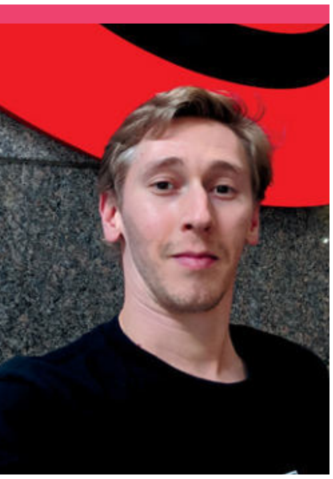
A great 'back to basics' video course which takes you through all the steps of installing and using SQLite. Over 90 minutes of video in bite-size pieces.
magpi.cc/sqlitepythonudemy

USING MYSQL DATABASES WITH PYTHON

Just like for SQLite, Udemy offers a full nothing-to-something course on using MySQL with Python. The two hours of video usefully cover installation of the server.
magpi.cc/mysqlpythonudemy

MONGODB AND PYTHON: QUICK START

Crucially, this two-hour video course covers when to choose a NoSQL (or document) database and the key differences between this and traditional SQL engines.
magpi.cc/mongodbpythonudemy



Ben Nuttall

We catch up with the Raspberry Pi Foundation's fourth employee, and one of the creators of Raspberry Jams, who now works for the BBC in an innovation team called BBC News Labs

- Day job **Software Engineer**
- Community role **Ex-Community Manager**
- Website **bennuttall.com**
- Twitter **@ben_nuttall**

Have you ever used GPIO Zero? Installed Python libraries from pip? Visited the Raspberry Pi website? Read Raspberry Pi documentation? Been to a Raspberry Jam? Read the events calendar pages in this very magazine? Then you've used something that the legendary Ben Nuttall has worked on.

After six years working for the Raspberry Pi Foundation, Ben has moved on to work for the BBC, so we wanted to celebrate his role in helping to build the community from day one.

"Throughout my final year at university (2010/11), I started getting into Linux and began learning Python," Ben recalls. "I heard about the Raspberry Pi project in its infancy – back then it was just a claim that they were going to make a small Linux computer for $25, which sounded incredible and very much relevant to my interests."

Like many people on 29 February 2012, he got up bright and early to try to order one of the first run of Raspberry Pi computers. He managed to place an order, but had to wait until June to get his hands on one. That didn't dissuade him from using Raspberry Pi, though, and the rest is history.

What was your community role before joining Raspberry Pi?

As soon as my Raspberry Pi arrived, I installed the OS and booted it up. I had no idea how to get to the Desktop, and all I could think to do with it was to type simple commands into the Python shell. I knew I wasn't making the most of it, so I thought I should find a community group to learn from. Back then I was attending several programming user groups at the local tech space in Manchester, so I asked them if there was going to be a Raspberry Pi group. They said I should start one – so I did. I called it Manchester Raspberry Jam, and other people were setting up their own around the same time – notably Alan O'Donohoe in Preston. Raspberry Jams naturally became

The Birthday Jams are part of the evolution of Raspberry Jams!

Ben could be found promoting Raspberry Pi at many events around the globe

▲ One of Ben's main aims has been to encourage more young people into tech and digital making

▲ Presenting at one of the earliest Raspberry Jams

> We've since trained over 1000 teachers through the programme

a network of similar events around the world.

I started Pi Weekly, an email newsletter covering news, projects, and articles from the Raspberry Pi community. This got picked up by Liz [Upton] on the Raspberry Pi blog, which grew the readership massively overnight. The Manchester Jam ran monthly, and I also ran some workshops for teachers and kids at the Museum of Science and Industry as part of my role as a STEM Ambassador.

How did you get hired by Raspberry Pi?

Liz had a keen eye on all the great things going on in the community, so she'd noticed all the stuff I was doing. One day she emailed me out of the blue saying they were looking to hire someone to redevelop the website and do some outreach – and would I be interested? Obviously I was, and I made plans to relocate to Cambridge, where I joined as the Foundation's employee #4.

What work did you do with Raspberry Pi?

My main task initially was to revamp the website. A month after I started, Carrie Anne Philbin and Dave Honess joined the team and, along with Clive Beale and Liz, we worked to launch a new site with learning resources, documentation, and more, keeping the existing blog intact. On 1 April we launched a joke website (green text on black, monospace font – it was great!) and on 2 April we launched the real one. I'll never forget the commenter who expressed his hatred for the real site's design, suggesting we go back to the green-screen one.

Then we launched Picademy, Carrie Anne's brainchild: a free professional development course for teachers. We've since trained over 1000 teachers through the programme, and it's honestly one of the most rewarding things I've worked on.

My first year was hectic: I spent countless hours writing up pieces of documentation (which I still refer to myself to this day), coming up with ideas for learning resources, adding new content to the website, doing Picademy and other workshops, and writing Pi Weekly every Friday. In the summer I did a three-week driving tour of the US, visiting schools, universities, and hackspaces. That was a lot of fun!

I went on to become Community Manager, looking after Raspberry Jams and the wider Raspberry Pi community, and then Technical Programme Manager, running the technical aspects of Astro Pi, which means doing sysadmin for the Raspberry Pi computers on the International Space Station.

Legacy of piwheels

"Piwheels saves people countless hours installing libraries. I see people build amazing projects with GPIO Zero, as it helps them progress from the basics to more advanced electronics and code. I see the piwheels Twitter bot (**@piwheels**) tweeting the stats showing how many downloads, and how much time has been saved. At the time of writing, piwheels has over 16 million downloads, saving almost 200 years of build time."

This Month in Raspberry Pi

Big birthday list!

Here are just some of the Raspberry Pi events celebrating its eighth birthday

As we mentioned last issue, Raspberry Pi is turning eight this year! Or two, if you only count leap years. Either way, in honour of this birthday, any Raspberry Jams taking place between Saturday 15 February and Sunday 15 March can be used to celebrate Raspberry Pi!

Here are the events that have been planned by the time we go to print. For a more up-to-date list, head to **rpf.io/jam**.

UK

Barnstaple Raspberry Jam
Barnstaple

BLC Raspberry Jam
Buxton

CamJam
Cambridge

Chelmsford Raspberry Jam
Chelmsford

Chi Raspberry Jam
Chichester

Exeter Raspberry Jam
Exeter

Gateshead Raspberry Jam
Gateshead

Huddersfield Raspberry Jam
Huddersfield

Hull Raspberry Jam
Hull

Leeds Raspberry Jam
Leeds

Leicester Hackspace Raspberry Pi Birthday Jam
Leicester

London Raspberry Pint
London

Margate Raspberry Jam
Margate

Beeston Raspberry Jam
Nottingham

South Devon Tech Jam
Paignton

Cornwall Tech Jam
Redruth

Sheffield Raspberry Jam
Sheffield

Southend Raspberry Jam
Southend-on-Sea

York Big Birthday Pi Jam
York

USA

Raspberry Pi Birthday Jam @ California High School
California

Fresno Ideaworks Pi Day Party
California

Berkeley Raspberry Pi Jam
California

Exeter Raspberry Jam
California

South San Francisco Public Library Raspberry Jam
California

Raspberry Jam @ Denver Library
Colorado

Raspberry Jam Catoosa
Georgia

Jelly's Raspberry Jam Party
Maryland

Raspberry Pi Big Birthday Jam Ann Arbor
Michigan

Triangle Area Makers Raspberry Jam!
North Carolina

Hunterdon County Library Raspberry Jam
New Jersey

Los Alamos Raspberry Jam
New Mexico

Long Beach Public Library Raspberry Jam
New York

Rochester Makerspace Raspberry Jam
New York

TEC Raspberry Jam
Tennessee

Raspberry Pi 8th Anniversary Celebration!
Utah

Roanoke Raspberry Jam
Virginia

Columbia Basin Jam
Washington

Key Tech Pi Day Jam
Washington

West Sound CoderDojo Raspberry Jam
Washington

Raspberry Jam WV
West Virginia

Europe

Raspberry Jam Zelzate
Belgium

Varna Raspberry Jam
Bulgaria

Raspberry Jam des inspirés 2020
France

Raspberry Jam de la Licorne
France

Pi and More 12½
Germany

Votanikos Jam
Greece

Torino Raspberry Pi Jam
Italy

RASPBERRY PiZZA Jam
Italy

AmsterJam
Netherlands

CoderDojo Baarle Raspberry Birthday Party
Netherlands

Hacker Jam Cluj Raspberry Pi 8th Birthday
Romania

Raspberry Jam Russafa
Spain

PiWars Turkey
Turkey

İstanbul Raspberry Pi Jam Etkinliği 2020
Turkey

Rest of the World

Litoral Raspberry Jam
Argentina

Raspberry Pi JAM Buenos Aires
Argentina

Santiago del Estero Raspberry Jam
Argentina

Inverell Raspberry Jam
Australia

Perth Raspberry Pi Jam & Workshop
Australia

Raspberry Jam on Broadway
Australia

Bangladesh Raspberry Jam
Bangladesh

Calango Raspberry Jam
Brazil

Raspberry Jam São Paulo
Brazil

Raspberry Jam Halifax
Canada

Pi Jammin 2020
Canada

Ottawa Jam
Canada

Shenzhen Raspberry Jam
China

Camagüey Raspberry Pi Jam 2020
Cuba

Pi Jam @ PiLabs
India

Raspberry Pi Jam Kochi
India

Abhinav Raspberry Jam
India

Atria Labs Raspberry Jam
India

Bhopal Raspberry Jam
India

PiWorld Raspberry Pi Jam
India

ResPro Labs Raspberry Pi Jam
India

Pi Jam 2020 @ SRiX Warangal
India

Raspberry Jam NextTech
India

Raspberry Pi Jam by ILUG-D & HHC-D
India

Raspberry Jam DU
India

Oscillations Raspberry Jam
India

BVP Raspberry Jam
India

Trivandrum Raspberry Jam Big Birthday Weekend
India

Raspberry JAM BIG Birthday Weekend @Surat Raspberry Jam
India

DTC Technical Raspberry Jam
India

Next Tech Lab AP Raspberry JAM
India

UI Raspberry Jam Iran
Iran

Itierio Boys' High School Raspberry Jam
Kenya

Perak Technology Academy: Raspberry Pi Jam
Malaysia

Jam Kathmandu
Nepal

Ikeja Tech Mates Raspberry Jam
Nigeria

Pi Day with Free City Hacks
Nigeria

Sargodha Raspberry Jam
Pakistan

Raspberry Jam LIMA
Peru

Davao Raspberry Birthday Jam
Philippines

Raspberry Jam Lisboa
Portugal

Suñu Raspberry Jam
Senegal

3rd Annual Cape Town Raspberry Jam/Makeathon/Hackfest!!
South Africa

Raspberry Pi Jam in Taipei 2020
Taiwan

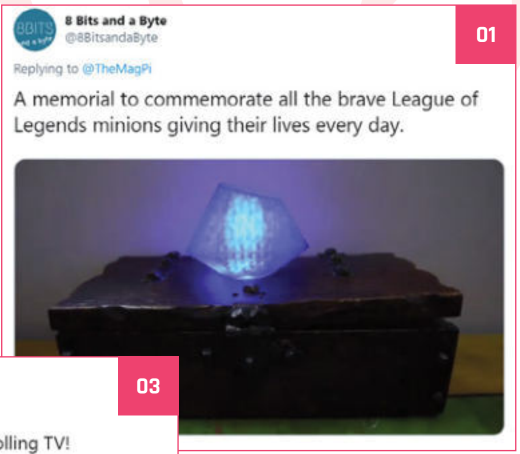
Bangkok Raspberry Jam
Thailand

Dubai Raspberry Birthday Jam
United Arab Emirates

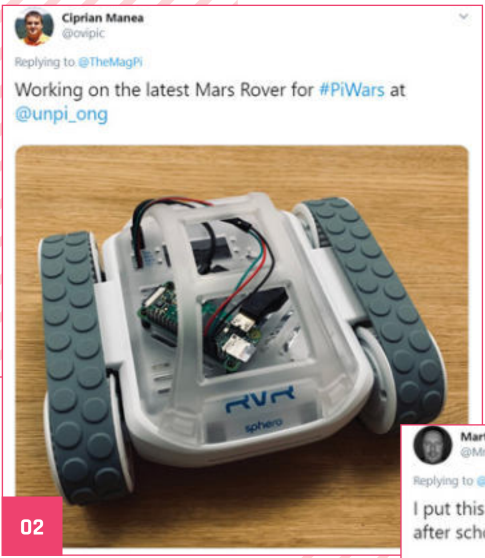
MagPi Monday

Amazing projects direct from our Twitter!

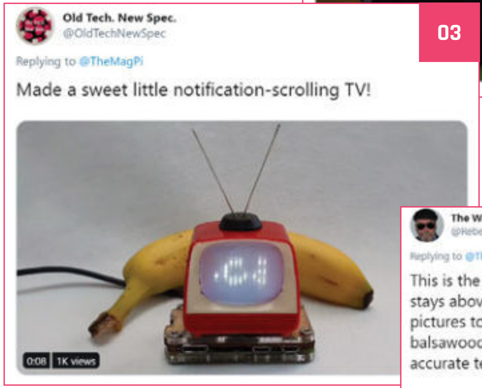
Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they've made. Here is a small fraction of them. Follow along at the hashtag #MagPiMonday.


01

8 Bits and a Byte
@8BitsandaByte
Replying to @TheMagPi
A memorial to commemorate all the brave League of Legends minions giving their lives every day.

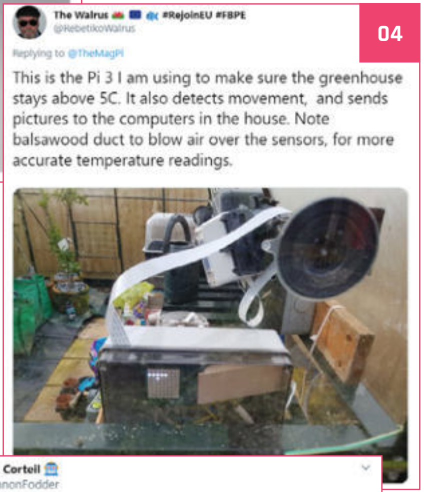

02

Ciprian Manea
@ovipic
Replying to @TheMagPi
Working on the latest Mars Rover for #PiWars at @unpi_ong

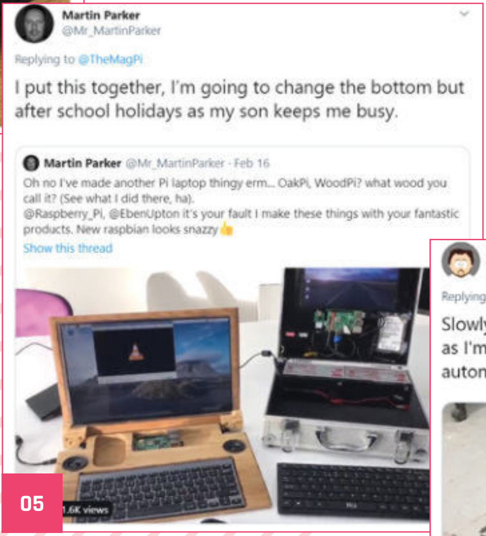

03

Old Tech. New Spec.
@OldTechNewSpec
Replying to @TheMagPi
Made a sweet little notification-scrolling TV!

0:08 1K views


04

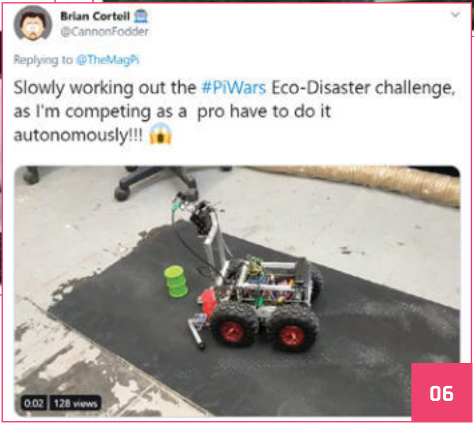
The Walrus #RejoinEU #FBPE
@RebetikoWalrus
Replying to @TheMagPi
This is the Pi 3 I am using to make sure the greenhouse stays above 5C. It also detects movement, and sends pictures to the computers in the house. Note balsawood duct to blow air over the sensors, for more accurate temperature readings.


05

Martin Parker
@Mr_MartinParker
Replying to @TheMagPi
I put this together, I'm going to change the bottom but after school holidays as my son keeps me busy.

Martin Parker @Mr_MartinParker · Feb 16
Oh no I've made another Pi laptop thingy erm... OakPi, WoodPi? what wood you call it? (See what I did there, ha).
@Raspberry_Pi, @EbenUpton it's your fault I make these things with your fantastic products. New raspbian looks snazzy👍
Show this thread

1.6K views


06

Brian Corteil
@CannonFodder
Replying to @TheMagPi
Slowly working out the #PiWars Eco-Disaster challenge, as I'm competing as a pro have to do it autonomously!!! 😱

0:02 128 views

01. RIP to some real ones

02. We love the aesthetics of this Mars Rover

03. The banana is to scale, and this little TV is amazing

04. A very useful device, especially during the strong weather in the UK lately

05. We love these DIY laptops!

06. Brian is always creating robots that wow us

Best of the rest!

Here are some other great things we saw this month

SMART BELT

"Built a smartbelt using a Raspberry Pi," begins this Reddit post. "Is it useful? No. But is it PRACTICAL? Also, no." We love it anyway.

▶ **magpi.cc/smartbelt**

DNA SCULPTURE

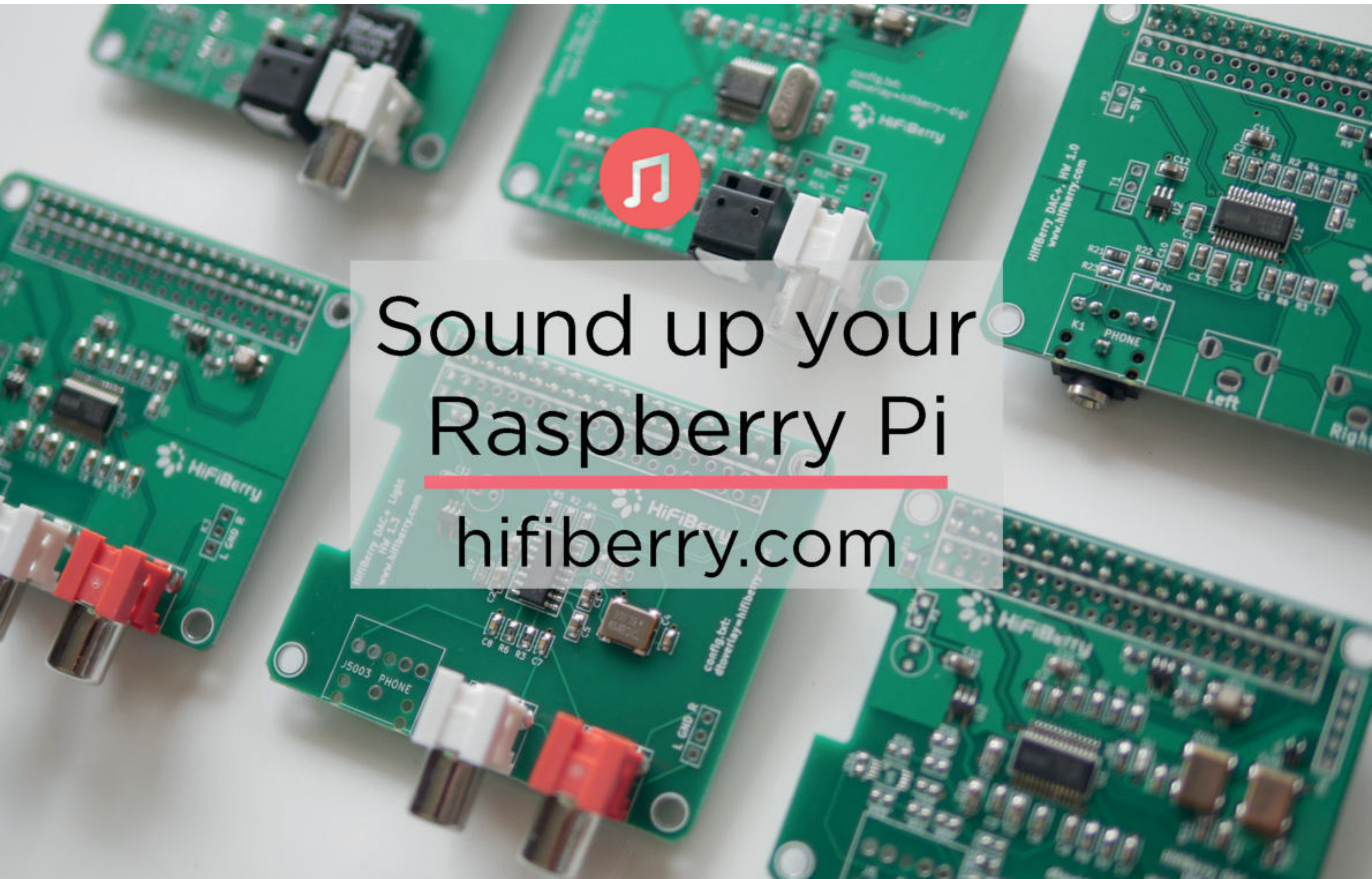
This cool, 3D-printed sculpture uses colours to display the base pairs of the creator's own DNA.

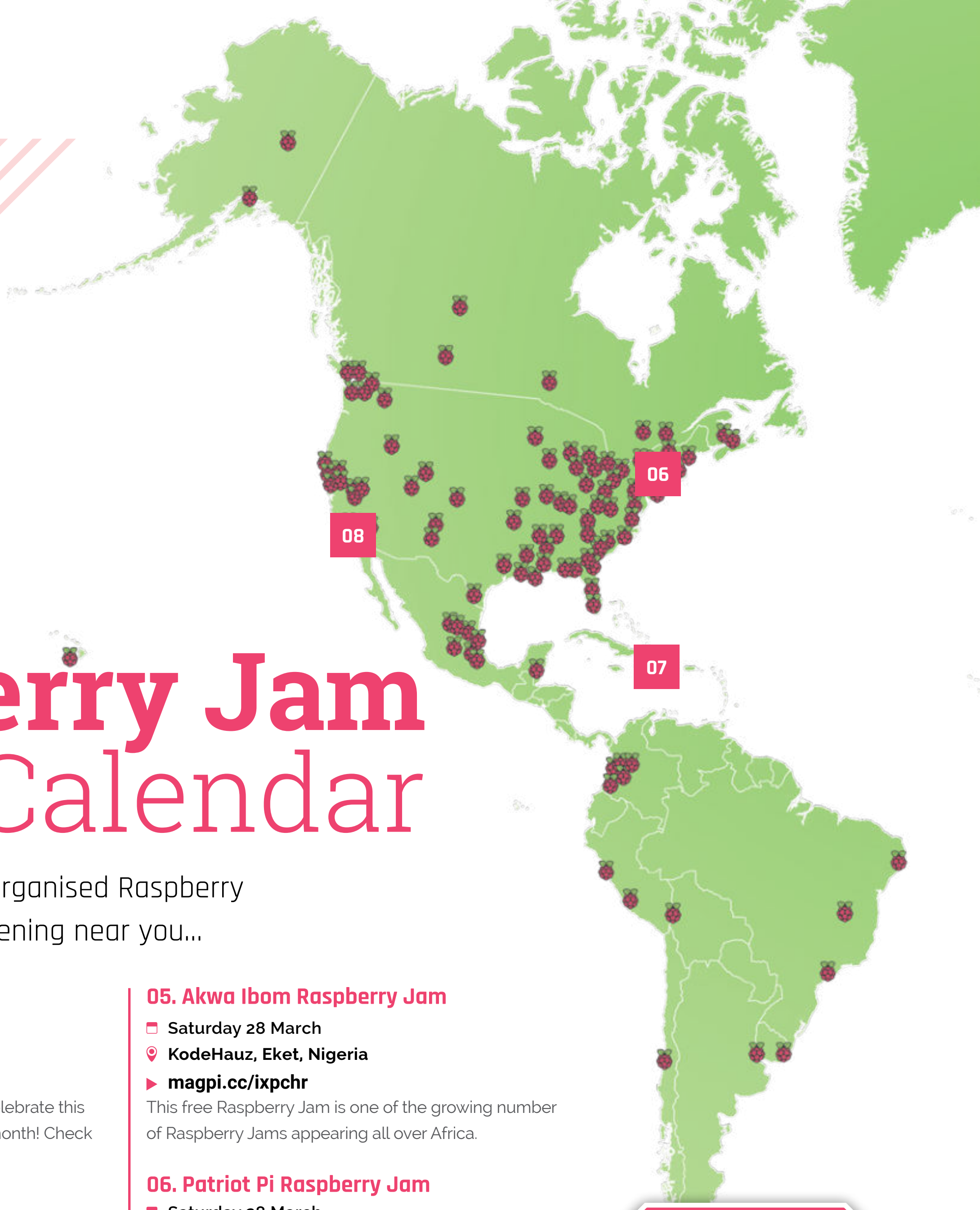
▶ **magpi.cc/genome**

CROWDFUNDING A PROJECT?

If you've launched a Raspberry Pi-related project, let us know!

magpi@raspberrypi.org





Raspberry Jam Event Calendar

Find out what community-organised Raspberry Pi-themed events are happening near you...

01. Birthday Jams!

Saturday 29 February

Earth, Sol System, Milky Way

rpf.io/jam

Raspberry Pi turns eight this year, and to celebrate this there will be Jams around the world for a month! Check page 84 for more...

02. Raspberry Jam Malvern Hills Student Edition

Wednesday 18 March

Wyche Innovation Centre, Malvern, UK

magpi.cc/jpkv9y

Free monthly after-school workshops for students interested in software coding and hardware development.

03. Raspberry Jam Zelzate

Saturday 21 March

Openbare Bibliotheek Zelzate, Zelzate, Belgium

magpi.cc/xnqidt

Everyone is welcome to start, share, and work on their own project(s) in a fun and relaxed atmosphere.

04. Topsham Raspberry Jam

Saturday 21 March

Nancy Potter House, Topsham, UK

magpi.cc/ysrxpx

A Code Club turned Jam, you'll need to call the number at the League of Friends website to enrol.

05. Akwa Ibom Raspberry Jam

Saturday 28 March

KodeHauz, Eket, Nigeria

magpi.cc/ixpchr

This free Raspberry Jam is one of the growing number of Raspberry Jams appearing all over Africa.

06. Patriot Pi Raspberry Jam

Saturday 28 March

Cherry Valley-Springfield Central School, Cherry Valley, NY, USA

magpi.cc/bedmuy

A fun get-together where you can try out Raspberry Pi, meet others, and see some cool projects.

07. Raspberry Pi Santo Domingo

Saturday 28 March

Esmeralda Guest House, Santo Domingo, Dominican Republic

magpi.cc/y3erwe

An event open to experts to show off their projects, and newcomers to find out about Raspberry Pi.

08. 2020 Saddleback Valley Raspberry Jam

Saturday 28 March

Laguna Hills High School, Laguna Hills, CA, USA

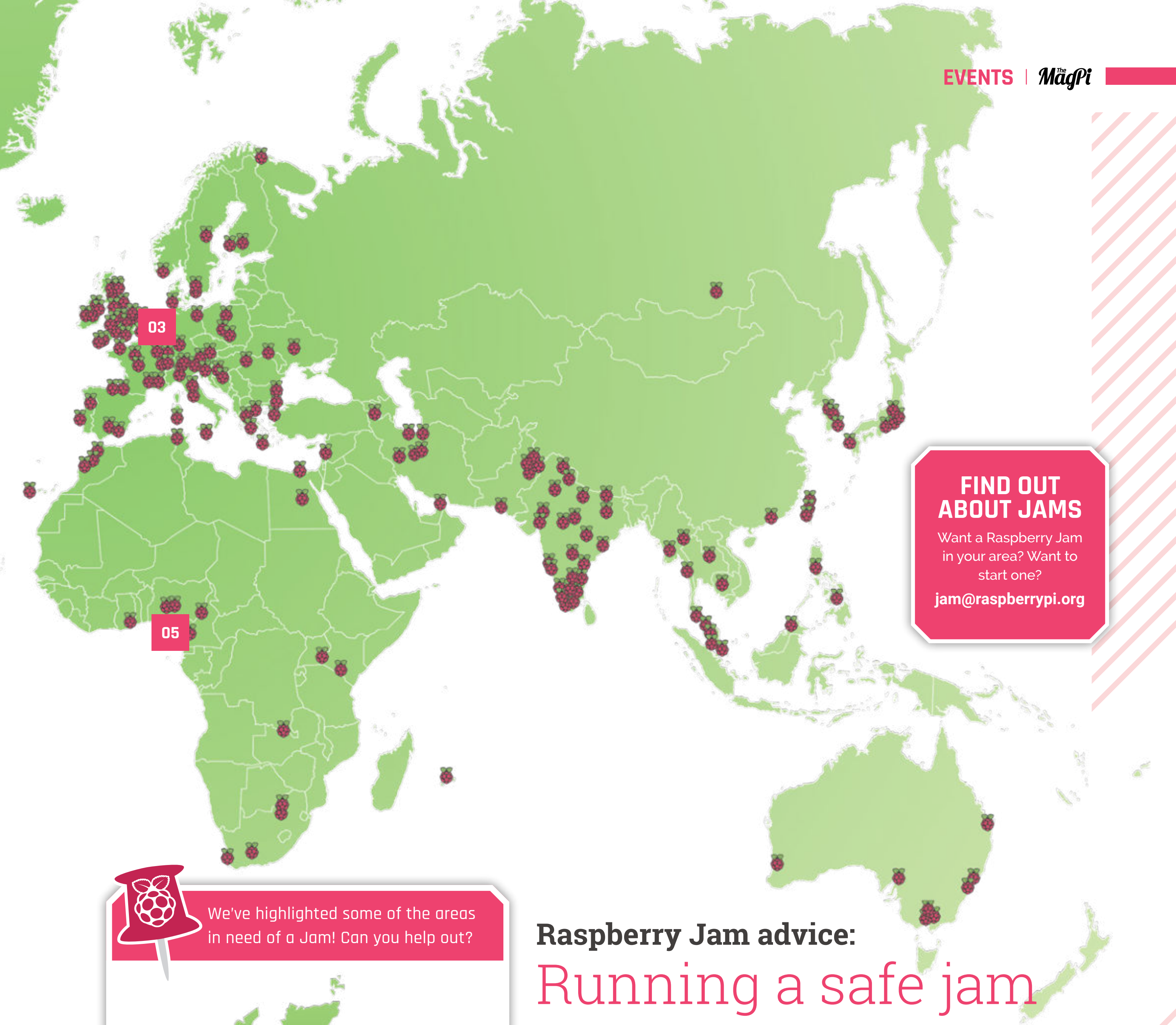
magpi.cc/yttwks

Projects will be on display that community members have created, as well as beginner workshops on Raspberry Pi.

FULL CALENDAR

Get a full list of upcoming events for March and beyond here:

rpf.io/jam

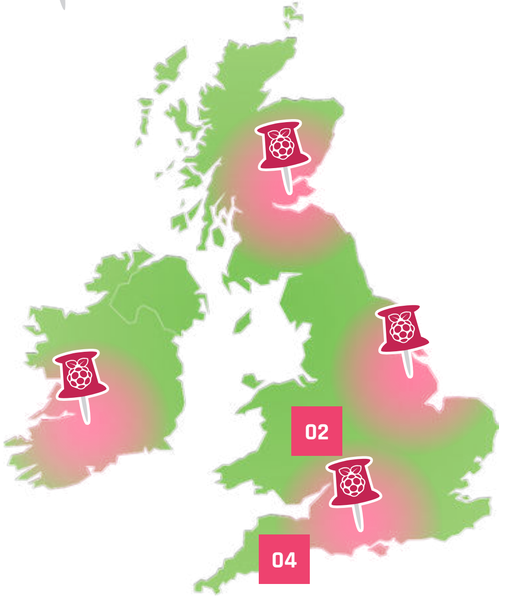


FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area? Want to start one?

jam@raspberrypi.org

We've highlighted some of the areas in need of a Jam! Can you help out?



Raspberry Jam advice:

Running a safe jam

"Start every event by explaining where the fire exits are, and where people should assemble in the event of a fire alarm. Talk to your volunteers and discuss how you would help a wheelchair user evacuate. Usually the plan is that someone stays with them to help them evacuate last, so they get a clear path and other people don't trip over them. Some buildings have a wheelchair refuge area with a communication panel: make sure you know where it is."

Andrew Oakley – Cotswold Jam

Every Raspberry Jam is entitled to apply for a Jam starter kit, which includes magazine issues, printed worksheets, stickers, flyers, and more. Get the book here: **rpf.io/guidebook**

Your Letters

USA subs

I see there is a new subscription offer for people in the US! $60 for a year sounds like an amazing price. How long is it lasting for? Is it for the whole of North America?

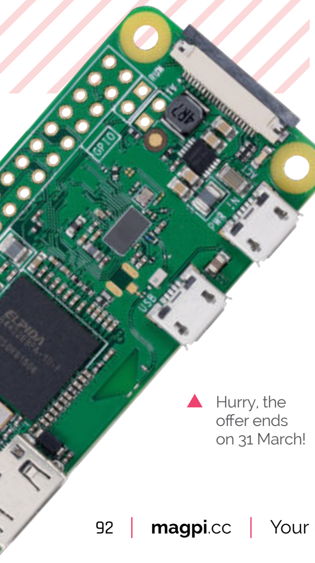
Also, are there any free gifts with this subscription? I have been eyeing up the older version that came with a Raspberry Pi Zero W, but it was a little out of my price range.

Liza via email

Our latest American subscription offer is running until the end of March: $60 for an entire year! Unfortunately, it's currently only for the USA, although HackSpace magazine is also doing a similar offer until the end of March!

Like our usual yearly subscription, you get a Raspberry Pi Zero W, a case for it with a selection of covers, and a small set of converter cables for the smaller ports.

You can find out about the new US subs for *The MagPi* at **magpi.cc/usa**, or for HackSpace magazine at **hsmag.cc/usa.**



Hurry, the offer ends on 31 March!

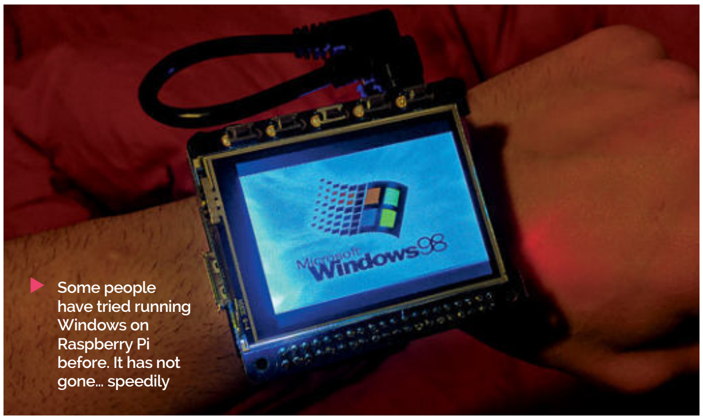
x86 now?

With the upgrade to Raspberry Pi 4 and all the firmware updates I've read about, I was wondering if it was finally powerful enough to run x86 operating systems and software? I keep thinking about a portable After Effects computer, and it would be amazing if it runs on Raspberry Pi.

Dom via email

Unfortunately, even with all the extra power a Raspberry Pi 4 possesses, it's still based on an ARM processor, so running x86 operating systems and software is not something you want to be doing with a Raspberry Pi.

Also, After Effects takes up a massive amount of resources – we've had powerful x86 PCs that have struggled with it at times, so a Raspberry Pi 4 is perhaps not what you'd be wanting. You'll probably need to get a laptop suited for the job.



Some people have tried running Windows on Raspberry Pi before. It has not gone… speedily

Just an LED

Every week I see you asking for people to show off their projects from the weekend, and I love seeing them. However, I feel like all my very simple projects aren't worth sending to you. Would you like to see my LED circuit?

Nate via Twitter

Everyone has to start from somewhere. A lot of the makers in the community have been working on projects for years, and many even have tons of experience in coding or engineering because of their day job.

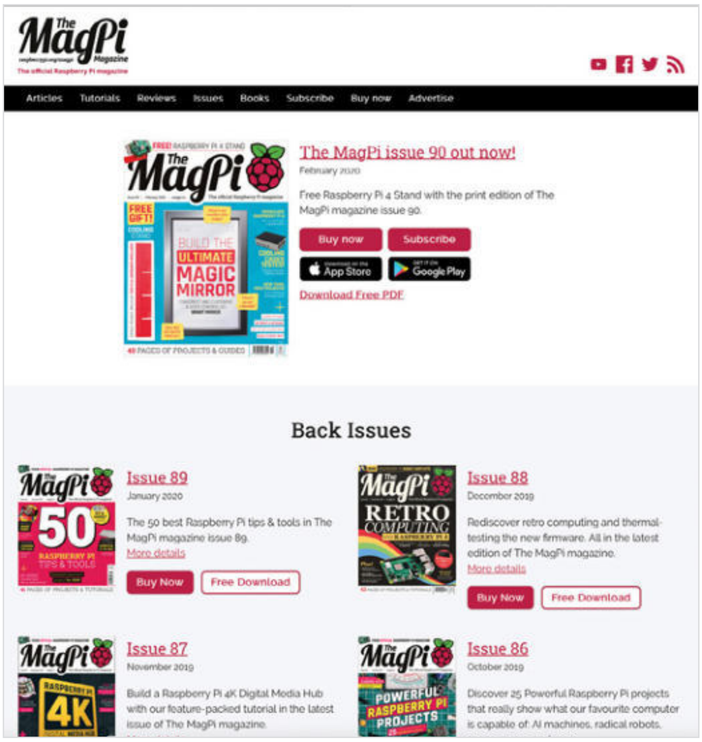
However, you should never let that put you off making what you can, or want. It's not a competition, and as long as you're having fun or getting something else out of making your project, then that makes it valid.

This is part of the message we want to promote during #MonthOfMaking. Share away with your projects!

▼ We welcome any and all projects for #MonthOfMaking

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
Raspberry Pi 4 2GB upgrade!
Starter Electronics
#MonthOfMaking
Let's build together!
Top 10 Wearable Projects
40 PAGES OF PROJECTS & TUTORIALS

▼ Remember, our PDFs are free to download from our website if you need another way to stock your library!



The MagPi issue 90 out now!
February 2020
Free Raspberry Pi 4 Stand with the print edition of The MagPi magazine issue 90.
Download Free PDF

Back Issues

Issue 89
January 2020
The 50 best Raspberry Pi tips & tools in The MagPi magazine issue 89.
More details

Issue 88
December 2019
Rediscover retro computing and thermal-testing the new firmware. All in the latest edition of The MagPi magazine.
More details

Issue 87
November 2019
Build a Raspberry Pi 4K Digital Media Hub with our feature-packed tutorial in the latest issue of The MagPi magazine.

Issue 86
October 2019
Discover 25 Powerful Raspberry Pi projects that really show what our favourite computer is capable of: AI machines, radical robots.

MagPi for my town

Is it possible to get *The MagPi* at my local library or even sold at my book store? I think people would benefit from it being available!

Roman via Facebook

We don't control who stocks the magazine – your library will need to order the magazine; the same with your book store. We can't force them to do this, but if you mention it to your library and the shop's manager, they might be able to work something out!

Contact us!

Twitter	@TheMagPi
Facebook	magpi.cc/facebook
Email	magpi@raspberrypi.org
Online	raspberrypi.org/forums

WIN RASPBERRY PI DESKTOP KIT SIGNED BY EBEN UPTON!

The Raspberry Pi Desktop Kit comes with everything you need to start using your Raspberry Pi. This one is extra special, though: it has been signed by Eben Upton himself!

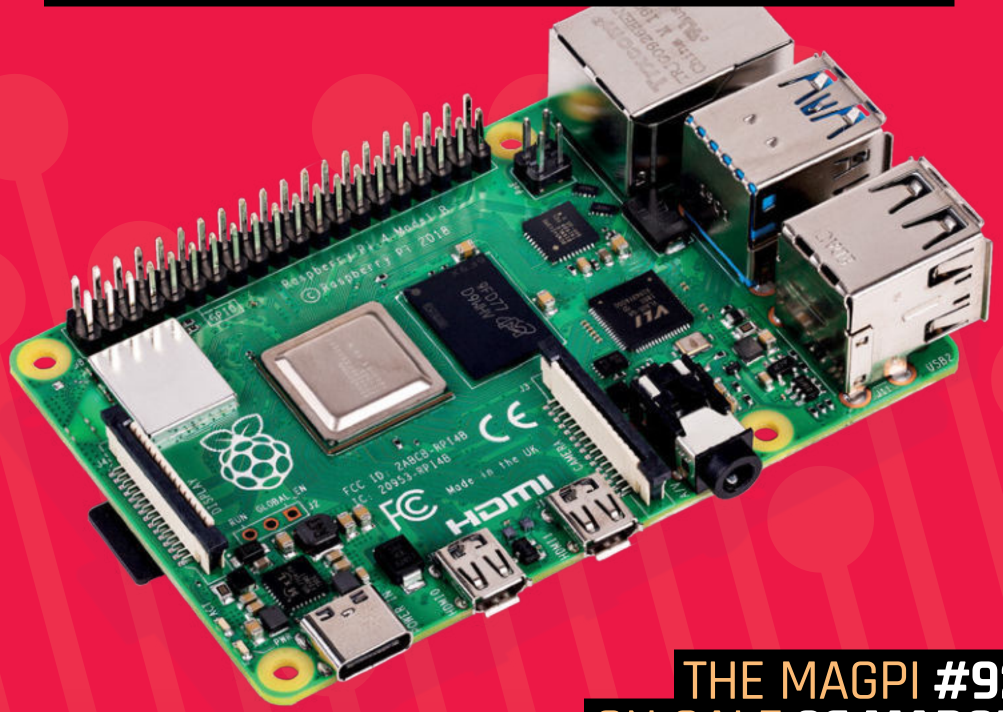
Head here to enter: magpi.cc/win | **Learn more: magpi.cc/desktopkit**

Terms & Conditions

Competition opens on **26 February 2020** and closes on **1 April 2020**. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram or Facebook.

RASPBERRY PI 4 TROUBLESHOOTING

DIAGNOSE AND FIX COMMON PROBLEMS



THE MAGPI #92
ON SALE 26 MARCH

Plus!

Set up a smart sourdough incubator

Weather & pollution monitoring

Build an Instagram clock

DON'T MISS OUT!
magpi.cc/subscribe

TWITTER	@TheMagPi
FACEBOOK	fb.com/MagPiMagazine
EMAIL	magpi@raspberrypi.org

EDITORIAL

Editor
Lucy Hattersley
lucy@raspberrypi.org

Features Editor
Rob Zwetsloot
rob.zwetsloot@raspberrypi.org

Sub Editors
Phil King and Nicola King

ADVERTISING
Charlotte Milligan
charlotte.milligan@raspberrypi.org
+44 (0)7725 368887

DESIGN
criticalmedia.co.uk

Head of Design
Lee Allen

Designer
Sam Ribbits, Harriet Knight

Illustrator
Sam Alder

CONTRIBUTORS
Mike Cook, David Crookes, PJ Evans, Gareth Halfacree, Rosemary Hattersley, Simon Long, Martin Mander, Simon Monk, Mark Vanstone

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.org

Director of Communications
Liz Upton

CEO
Eben Upton

DISTRIBUTION
Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS
Unit 6 The Enterprise Centre
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE
+44 (0)1293 312193
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk

Making for well-being

Martin Mander on the healing properties of tinkering on mental health

I'm thinking of getting a Raspberry Pi tattoo. Nothing too fancy, just the logo (given the right approvals), or maybe a 40-pin header. It's scary and would be my first, but it'd be a fitting expression of the depth of feeling I have for the little green board. It sounds a bit corny, but discovering Raspberry Pi and the culture around it changed my life.

Back in 2014, I'd been repurposing broken old tech for a while, but my projects always needed to be connected to something else, like the vintage phone with Google voice search that needed a laptop to work. Then a chance comment on one of my projects pointed me in the direction of Raspberry Pi and I was drawn in straight away, hatching a grand plan to put a large HD screen in an old VCR to make a standalone retro media centre. I'm gazing wistfully at it now – it mostly still works, though the final assembly was like sitting on a stuffed suitcase, so I don't dare reopen it to fix the sticky eject mechanism.

It was a genuine labour of love. Shortly after starting the VCR project, my wife Claire was diagnosed with breast cancer, and the family had a truly terrible year of waiting, worry, and coping with debilitating chemotherapy. What I remember throughout, though, was the value of having just a few minutes a day when I could pop to the garage and tinker with the Raspberry Pi project. Often all I got done was think a bit, or find two matching screws, but doing something creative, with an entire online community to turn to, made all the difference to my well-being. Just as you remember where you were when you last heard a song, every badly soldered switch and off-centre drill hole has its own story to tell of those days. We struggled, but the making helped me carry on and the family pushed through a dark winter together.

> "It mostly still works, though the final assembly was like sitting on a stuffed suitcase"

Back-seat maker

After a long but successful treatment, Claire's hair started to grow back in the spring, and interest grew in the finally complete VCR project. With every positive comment or mention, she reminded me that spray-painting it Raspberry red had been her sick-bed idea, and she should consequently have the credit – spawning a term we still use, 'back-seat maker'.

I was lucky enough to show the VCR at the Raspberry Pi 3rd Birthday event that year, a pretty lonely-looking project with a table to itself, but that day opened my eyes to the incredibly diverse and positive community that surrounds Raspberry Pi. I'm proud to be a part of it and am still in regular contact with people I met that day.

Endless inspiration

Since then, I've published a dozen projects, focusing on practical uses for obsolete tech that showcase the flexibility of Raspberry Pi. I thought the ideas would dry up eventually, but as more HATs appear, the boards evolve, and the community grows stronger, there's more to be inspired by in 2020 than ever before. I'm also constantly inspired by my wife's courage, and will always remember with gratitude how my own Raspberry Pi journey began. Now where can I find a slightly broken 1970s tattoo gun?

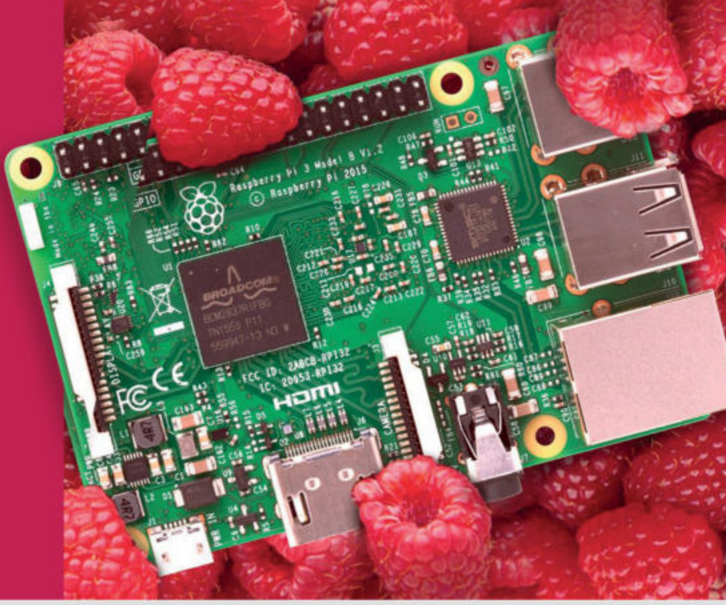
AUTHOR

Martin Mander

Martin has a passion for bringing new purpose to broken old technology with Raspberry Pi, and shares his projects on the 'Old Tech, New Spec' YouTube channel.

kyliemander.com

American **Raspberry Pi** Shop



- Displays
- HATs
- Sensors
- Cases
- Arcade
- Swag
- Project Kits
- Cameras
- Power Options
- Add-on Boards
- Cables and Connectors
- GPIO and Prototyping

Partner and official reseller for top Pi brands:

adafruit · sparkfun ELECTRONICS · PIMORONI · HiFiBerry · and many others!

Price, service, design, and logistics support for **VOLUME PROJECTS**

USA PiShop.us

Canada BuyaPi.ca

Raspberry Pi APPROVED RESELLER

PIPER

INSPIRE FUTURE GENERATIONS
WITH PIPER COMPUTER KIT 2

the award-winning computer kit for ages 8+

Find out more about this and many other products at **www.okdo.com**